MECANISMO DE INTEGRACIÓN DE PANDABOARD EN EL IDE DE ARDUINO EN UN CONTEXTO DE IOT

HAROLD ANDRES ADRADA GOMEZ
JORGE ARMANDO AGUIRRE REINA

Universidad del Cauca
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
GRUPO DE INGENIERÍA TELEMÁTICA
POPAYÁN, DICIEMBRE DE 2015

MECANISMO DE INTEGRACIÓN DE PANDABOARD EN EL IDE DE ARDUINO EN UN CONTEXTO DE IOT

HAROLD ANDRES ADRADA GOMEZ
JORGE ARMANDO AGUIRRE REINA

Monografía presentada para optar al título de Ingeniero en Electrónica y Telecomunicaciones

Director: Ing. Fulvio Yesid Vivas Cantero
Co Director: PhD. Ing. Gustavo Adolfo Ramírez González.

Universidad del Cauca
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
GRUPO DE INGENIERÍA TELEMÁTICA
POPAYÁN, DICIEMBRE DE 2015

Resumen

Este trabajo de investigación explica el proceso de selección de un mecanismo de integración de la tarjeta de desarrollo Pandaboard en el contexto del internet de las cosas. Inicialmente se tiene un marco teórico a partir del cual se obtienen los resultados e investigaciones de trabajos relacionados con la Pandaboard y también con Arduino, no solo con la tarjeta sino también con el ambiente de desarrollo. Luego se identifica el problema del uso de la tarjeta Pandaboard, la cual, a pesar de tener unas características hardware sobresaliente, aprovechable y extensible dentro del contexto de la IoT, no cuenta con una herramienta de desarrollo adecuada o cómoda para el usuario final. Frente a esto se plantean una serie de objetivos con el fin de resolver la pregunta de investigación, la cual está motivada por todas las posibilidades que ofrece dicha tarjeta para la interconexión de las aplicaciones disponibles en Arduino con la nube.

Para este desarrollo se define una metodología de trabajo a través de la cual se definen también unos alcances. Dentro del trabajo, y una vez resuelta esta parte, se definen tres tipos de mecanismos para integrar a la Pandaboard con el IDE de Arduino; estos mecanismos pasan por una serie de evaluaciones preparadas por los investigadores y el tutor, a fin de filtrar y seleccionar el más adecuado de ellos. Finalmente se procede a desarrollar dicho mecanismo pasando desde la selección e instalación de un sistema embebido adecuado, de la configuración de la tarjeta como accespoint hasta la ejecución de un programa desarrollado desde un pc.

La valoración de este proyecto de grado se realiza a través de la evaluación del mecanismo dentro de un caso de estudio, orientado hacia el contexto del internet de las cosas, en el cual, la tarjeta recibe una serie de datos desde una red de sensores como un servidor UDP para posteriormente enviarlos hacia la nube como un servidor TCP. Este desarrollo es realizado por un grupo de estudiantes de últimos semestres utilizando el entorno Arduino adaptado para soportar la Pandaboard, y los métodos convencionales de programación para esta tarjeta.

Se finaliza con las conclusiones, las cuales permiten identificar la carencia en la documentación y los trabajos relacionados con la Pandaboard. También muestran como Arduino logra mejorar el desempeño de los desarrolladores de la Pandaboard, con el uso de una interfaz más intuitiva en comparación con los comandos de Linux.

Abstract

The next research work it is based at the selections process of the Pandaboard integration mechanism in the context Internet of things. Initially it has a theoretical framework from which the results obtained and the research of related Jobs with the Pandaboard and the Arduino too, not only with the development card but also with the development environment. It then identifies the Pandaboard's use trouble, which, although their hardware features outstanding, usable and extensible inside IoT context, it does not have a development tool adequate or confortable for the final user. Deal with this it raises a number of objectives in order to solve the research question, which is motivated for all the possibilities that offer this card for the interworking of the applications available in Arduino with the cloud.

For this development is defined a working methodology through which defined scopes too. In this work, and once settled this part, defined three types of mechanisms to integrate the Pandaboard with the Arduino's IDE; this mechanisms pass for a series of evaluations prepared for the researchers and the tutor, for to filter and select the most suitable of them. Finally it proceeds to develop that mechanism going for sleection and instalation at suitable embedded system, of the card's configuration like accespoint until the execution of a program developed from a computer.

The assessment of this gradutaion Project is carried out through of the evaluation at this mechanism inside of a study case, oriented to the IoT context, in which, the development card receives a data from a sensor network like a UDP server for

later send them to the cloud like a TCP server. This development is done by a group of students in their final semesters using the Arduino environment adapated for support the Pandaboard, and the conventional programming methods for this development card.

It ends with conclusions, which can identify the lack of documentation and work related Pandaboard. Arduino also show that manages to improve the performance of developers Pandaboard, using a more intuitive interface compared to Linux commands.

Tabla de contenido

Lista de Tablas

Lista de Figuras

Capítulo 1

1 Introducción.

1.1 Contexto General

Ideas como que objetos cotidianos del hogar y del entorno tengan capacidad para realizar labores de procesamiento son ya una realidad a través de la Internet de las Cosas (Internet of Things, IoT), la cual, contempla una red de objetos físicos con acceso a través de la Internet [1]. Cada objeto debería tener un identificador con un funcionamiento bastante simple, pero en términos de aplicabilidad no es tan sencillo; debido a ello, la proyección sobre esta tecnología es realmente amplia en tipos y en número de dispositivos que se podrían cubrir y vincular, a fin de facilitar el enlace entre el mundo real y las herramientas del mundo virtual [1]. La IoT está conectando nuevos espacios, como plantas de fabricación, redes de energía, centros de salud, centros comerciales, sistemas de transporte a Internet y sobre una múltiple variedad de campos; cuando un objeto puede representarse digitalmente, este puede ser controlado desde cualquier parte por un operario o un sistema inteligente, esta conectividad significa más datos recopilados de más lugares, con más formas de aumentar la eficiencia y la seguridad. [2]

Paralelo a esto, las tecnologías en el ámbito de los sistemas embebidos (vinculadas en muchos aspectos a la IoT) también están alcanzando un gran auge, a nivel software y Hardware aparecen más y mejores opciones para el uso e implementación de recursos orientados a este entorno, además de nuevos conceptos como los de Hardware abierto (Open Hardware), de igual manera, están creciendo muchas comunidades dedicadas al estudio y desarrollo de aplicaciones dentro de este contexto, no sólo en la parte empresarial sino también en la parte académica.

La mayoría de estas aplicaciones están desarrolladas bajo el concepto de Open Hardware, el cual, toma las mismas ideas del software libre para aplicarlas en su

campo, en lo referente a las cuatro libertades: libertad de uso, de estudio y modificación, de distribución, y de redistribución de las versiones modificadas [3]. El objetivo básico es la creación de diseños de tarjetas electrónicas de forma abierta, de manera que todas las personas que quieran puedan acceder, como mínimo, a los planos de diseño de los dispositivos, además, el Open Hardware presenta características tales como: la protección de la soberanía de cada país para evitar la dependencia de desarrollo extranjero; impulso para el desarrollo de hardware de calidad, abierto y económico; la reutilización y adaptación de diseños a nivel mundial de forma colaborativa; la promulgación de comunidades de desarrollo crecientes y participativas, evitando la alianza de computación confiable (trusted computing), y la gestión digital de derechos (DRM). [3]

Arduino es una comunidad de prototipado rápido y software embebido ampliamente conocido ya que ha presentado una serie de herramientas muy accesibles en el sentido de acercamiento hacia la tecnología, permitiendo que cualquier persona sin muchos conocimientos sobre entornos de programación y hardware pueda realizar sus propios proyectos [4], a su vez la Pandaboard es un hardware con unas características muy poderosas en cuanto a capacidad de procesamiento, que amplían las posibilidades de manejo, utilización e implementación en sistemas más grandes y sofisticados, tal es así, que en principio fue concebido para poder funcionar como un computador de bajo costo dadas sus características hardware, permitiendo conectar todos los periféricos que tiene un computador de escritorio y manejarlos a través de un sistema operativo cargado en una memoria de almacenamiento SD. [5]. Por consiguiente este trabajo se centra en el IDE Arduino y en el hardware Pandaboard.

1.2 Declaración del problema.

Pese a todas las ventajas y características de la Pandaboard, esta tiene una dificultad en cuanto a su programación desde el punto de vista comparativo frente a otras tarjetas que incluso son menos poderosas; la necesidad de demasiadas

herramientas software para su programación (Python, C, C++, gdb, binutils, etc.), esto eleva el grado de complejidad en el sentido de que se requiere usar varios lenguajes a nivel de sistema operativo y no de usuario, además, es necesario asegurar su completa compatibilidad a fin de garantizar el correcto funcionamiento de los códigos a implementar y posteriormente ejecutar.

De aquí, teniendo en cuenta el gran potencial de la tarjeta de desarrollo Pandaboard, la facilidad para programar en el IDE de Arduino y su utilidad dentro del contexto del Internet de las cosas surge la siguiente pregunta de investigación:

¿Cómo facilitar el desarrollo de aplicaciones embebidas usando la tarjeta Pandaboard en el contexto del IoT?

1.3 Estado del Arte.

En este capítulo se recopilan los conceptos y tecnologías en los que se cimienta este trabajo de grado y se presentan las experiencias de otros investigadores que constituyen el conocimiento base de la temática de investigación y un marco para identificar los posibles enfoques que pueden ser utilizados, las limitaciones que existen y los campos por explorar identificando los posibles aportes.

1.3.1 Trabajos Relacionados.

En la actualidad, existe un número importante de trabajos cuyos objetivos están enfocados a la implementación de prototipos y proyectos en los que se tiene como base la utilización de la plataforma Arduino, esto se debe a la gran simplicidad del proceso de programación y despliegue de servicios; es posible evidenciar, que muchos de estos proyectos están enfocados al uso de la Internet de las cosas IoT.

También se encontraron proyectos donde se trabaja sobre la plataforma Pandaboard, en estos se logra hacer uso del gran potencial que tiene esta tarjeta y se puede analizar su gran campo de aplicación.

En [6] se hace un manejo de sensores de proximidad para la localización de objetivos, en [7] se crea un prototipo para redes submarinas, en [8] se crea un método para la detección de un patrón ocular y la implementación del mismo en la librería de OpenCV; en estos proyectos se utilizan computadoras embebidas, esto a través del uso de la Pandaboard.

En [9] se presenta un sistema que notifica a un usuario sobre actividades inusuales en su hogar y los almacena para posterior acceso. El sistema utiliza la Pandaboard específicamente para el procesamiento de imágenes (Con el uso de librerías de OpenCV). En [10] se hace uso de tecnologías para el manejo inteligente de automóviles de manera autónoma en un vehículo miniatura, se utiliza un sistema en tiempo real para la interfaz software/hardware y para correr la aplicación una arquitectura ARM basada en Pandaboard y Razorboard.

Por otra parte, en [11] se encuentra una primera aproximación entre las plataformas Pandaboard y Arduino, en ese proyecto se hace un Robot que mejora la función de la persona encargada de recoger bolas en una cancha de tenis, la Pandaboard es usada como procesador principal responsable de las tareas intensivas y de las comunicaciones externas, la placa Arduino es usada para manejo de motores de baja potencia y para la interfaz de sensores, la comunicación entre la placa Arduino y la Pandaboard se hace vía puerto serial RS232.

En [12] se describe los detalles de la portabilidad (porting) de Linux a una board con ARM. La construcción del kernel de Linux, u-boot y el x-loader se describen en detalle, además de la descripción de la configuración de la SD.

En [2] se trabaja con la creación de librerías y sketchs para el manejo de un shield de telefónica, el objetivo es que trabaje tanto con la tarjeta Arduino como con su IDE, para ello, a lo largo del proyecto se hace uso de herramientas de manejo de versiones como Subversion (repositorio de pruebas y librerías) y plataformas web para realizar las labores de compilación continua, tal como Jenkins (sistema de integración continua), todo esto para efectos de garantizar un buen funcionamiento de las librerías y una completa integración de las herramientas del IDE con el Shield a usar.

1.3.2 Brechas del conocimiento.

Teniendo en cuenta el análisis realizado sobre los trabajos existentes, se logra evidenciar que la principal brecha es la falta de facilidades y accesibilidad a cualquier usuario sin conocimiento del proceso de programación en la Pandaboard y el limitado procesamiento del entorno Arduino en aplicaciones tan demandantes del IoT, es decir, no existe o no se evidencia de forma clara, un proyecto donde se trabaje sobre la utilización de este IDE sobre la Pandaboard, en la gran mayoría de los trabajos analizados se trabaja con las plataformas (Arduino o Pandaboard) pero de forma independiente.

En la tabla 1 se muestra las brechas existentes en cada uno de los trabajos ya descritos con anterioridad.

Trabajo	Brecha del Conocimiento
[6] [7] [8] [9] [10]	No se especifica el IDE de programación del respectivo HW de bajo costo para la implementación del robot que usa la Pandaboard.
[11]	Hacen uso de las placas Arduino y Pandaboard pero no se encuentran integradas; el uso de esta última viene dada como un backend de la información que se procesa tanto en esta, como en la tarjeta Arduino; el IDE de Arduino solo se usa para las boards Arduino mas no para la Pandaboard
[12]	Presenta una serie de procedimientos para el porting del kernel de Linux entre boards con arquitecturas ARM pero sin especificar necesariamente un trabajo sobre la Pandaboard
[2]	Desarrollan una serie de procedimientos y herramientas para la adaptación de un nuevo Hardware con la propiedad de ser manejado y editado desde el IDE de Arduino, lo hacen sobre la board propia pero no trabajan sobre la Pandaboard.

Tabla 1. Brechas Existentes.

1.4 Hipótesis.

Con base en la descripción del problema y el análisis del estado del arte, como hipótesis para el desarrollo de este trabajo de grado, se plantea:

Es posible adaptar el IDE de Arduino para soportar la tarjeta Pandaboard y así ampliar sus posibilidades de uso en el contexto de IoT.

1.5 Objetivos.

1.5.1 Objetivo general.

Proponer un mecanismo de integración de la Pandaboard en el IDE de Arduino para soporte en el contexto del IoT.

1.5.2 Objetivos específicos.

✓ Definir una serie de alternativas de integración de la Pandaboard y el Ambiente Arduino para soporte en el contexto del IoT.
✓ Implementar un mecanismo de integración entre la Pandaboard y el Ambiente de Arduino para soporte en el contexto del IoT.
✓ Validar mediante un caso de estudio el mecanismo de integración la Pandaboard en el Ambiente de Arduino para soporte en el contexto del IoT.

1.6 Metodología de Trabajo.

Con el propósito de cumplir los objetivos establecidos para el presente trabajo de grado, la metodología a seguir tiene como base inicial de referencia los lineamientos establecidos en el “Modelo Integral para el Profesional en Ingeniería” [13], por medio del cual, serán desarrolladas un conjunto de actividades. La metodología propuesta consiste en:

- El “Modelo para la Investigación Documental (MID)” [14]: empleado para la elaboración de una base de conocimiento teórica en temas como: Internet de la Cosas (IoT), Sistemas Embebidos, Mecanismos de Integración, entre otros.
- El “Modelo para la Construcción de Soluciones (MCS)” [15]: será empleado como metodología de referencia en el proceso de análisis, diseño, e

implementación de un prototipo de prueba, cabe aclarar que de todas las fases del MCS solo se consideran aquellas herramientas que permitan un desarrollo rápido de la solución.

1.7 Alcance.

El principal objetivo de este trabajo es proponer un mecanismo de integración de la Pandaboard en el IDE de Arduino para soporte en el contexto del IoT.

1.8 Contribuciones.

- ✓ Un informe de alternativas de integración de la Pandaboard en el IDE de Arduino para soporte en el contexto del IoT.
- ✓ Un mecanismo de integración de la Pandaboard en el IDE de Arduino para soporte en el contexto del IoT.
- ✓ Un caso de Estudio sobre un escenario de aplicación de ese mecanismo donde se evalúe la validación del mismo.
- ✓ Aporte a la comunidad de desarrolladores de Pandaboard, debido a que se logró crear una guía pionera de configuración de access point para la Pandaboard con el sistema Linux Debian embebido.
- ✓ Un artículo de divulgación donde se consignen los resultados del proyecto desarrollado.

1.9 Contenido de la monografía.

Partiendo de la metodología seleccionada que se utilizará y tendrá como referencia, a continuación son descritos los capítulos a realizar para alcanzar los objetivos propuestos.

El presente trabajo de grado contiene los siguientes capítulos:

Capítulo 1: INTRODUCCIÓN. Este capítulo presenta la definición del problema, el contexto general, estado del arte, se especifican los objetivos a alcanzar con el presente trabajo de grado, se determina la metodología a utilizar, el alcance y las contribuciones.

Capítulo 2: ESTADO ACTUAL DEL CONOCIMIENTO. En este capítulo se abordan las definiciones formales de conceptos y tecnologías claves en las que se fundamenta la integración de la Pandaboard en el IDE de Arduino en un contexto de IoT, además del estado actual del conocimiento relacionado con trabajos que se han adelantado sobre las diferentes áreas que abarca este proyecto, haciendo una recolección de la información relacionada con estos ítems que comprenden el marco de nuestro proyecto de investigación; al final se hace una selección de la información relevante para un estudio de mayor profundidad, determinando las características más importantes de cada concepto y área de conocimiento necesario para el proyecto de investigación.

Capítulo 3: OPCIONES DE INTEGRACIÓN ENTRE LA PANDABOARD Y EL IDE DE ARDUINO. Se describe el proceso de selección del modelo de integración más adecuado para la Pandaboard y el IDE de Arduino, además del modelado para el mecanismo de dicha integración; se detalla el proceso de selección de herramientas tecnológicas y metodológicas para la construcción e implementación del sistema, haciendo una evaluación del mecanismo para soporte de integración de la Pandaboard en el Ambiente de Arduino en el contexto del IoT. Dentro de la serie de alternativas consideradas se procede a realizar una evaluación para seleccionar la más afín y se comienza a estimar la efectividad del mecanismo de integración, una vez seleccionado el mecanismo se procede a justificar y soportar la elección realizada.

Capítulo 4: IMPLEMENTACIÓN DEL MECANISMO DE INTEGRACIÓN ENTRE LA PANDABOARD Y EL AMBIENTE DE ARDUINO. En este capítulo se desarrolla el proceso o mecanismo que logre hacer la integración de la Pandaboard dentro

del ambiente Arduino: Análisis de los resultados, síntesis y presentación de los mismos: estudio y valoración de las diferentes técnicas utilizadas en la construcción del mecanismo de integración implementado. En esta actividad, se realizan pruebas sobre los diferentes módulos del prototipo en un entorno de prueba, con el fin de comprobar que cumple con lo propuesto en este documento.

Capítulo 5: VALIDACIÓN MEDIANTE UN CASO DE ESTUDIO DEL MECANISMO DE INTEGRACIÓN ENTRE LA PANDABOARD Y EL AMBIENTE DE ARDUINO. En este capítulo se aborda la descripción detallada del trabajo realizado alrededor del desarrollo de la validación mediante un caso de estudio de la integración de la Pandaboard en el Ambiente de Arduino. Luego de la evaluación y realización de pruebas, se realizará la síntesis de información así como también de las conclusiones obtenidas a lo largo de todo el proceso, con el objetivo de fijar un precedente en este tipo de estudio, y sobre la cual se implementa la propuesta del proyecto; en la descripción se abarca la arquitectura definida para soportar la construcción del prototipo, los modelos de caso de uso del sistema, detalles de implementación y las pruebas realizadas para la evaluación del mismo, junto con los resultados obtenidos.

Capítulo 6: CONCLUSIONES, CONTRIBUCIONES Y TRABAJO FUTURO. Finalmente, se presentan las conclusiones a las cuales se llegó en el desarrollo del presente trabajo de grado, las principales contribuciones de la ejecución del proyecto y el planteamiento de diferentes ideas propuestas para la realización de trabajos futuros.

Capítulo 2

2 Conceptos Fundamentales.

2.1 Introducción.

En este capítulo se presenta una descripción detallada de las herramientas técnicas y metodológicas que soportan el proceso de construcción del Mecanismo de Integración. Para esto se describe la Metodología del Modelo para la Construcción de Soluciones [15], se hace una reseña de las tecnologías, de las herramientas hardware y de software que permiten el desarrollo de los objetivos del proyecto.

2.2 Metodología del Modelo para la Construcción de Soluciones.

Según [15] la metodología del modelo para la construcción de soluciones tiene como propósito: “construir una solución de calidad, oportuna y con costos competitivos y sobre todo que pretenda contribuir a la creación y enriquecimiento de la base de conocimiento/experiencia institucional”, debido al fuerte impacto positivo y al éxito que se tiene en el desarrollo de los trabajos de grado de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca, especialmente en el Departamento de Telemática se adopta como modelo para el desarrollo del presente trabajo.

En este modelo se destacan 5 pilares fundamentales nombrados por el autor de la metodología, y que se verán reflejados en el desarrollo del presente documento.

Para llevar a cabo cualquier proceso de desarrollo ingenieril se debe tener en cuenta:

- Ingeniería del sistema: Vista desde la calidad de un sistema para satisfacer necesidades y expectativas de su medio ambiente.
- Orientación a objetos: Utilizar el paradigma de orientación a objetos, destacando la importancia de diseñar componentes usando al máximo el reúso visto desde el proceso de desarrollo.
- Orientación a diseño: Implica el desarrollar sistemas/soluciones basados en una orientación que sea independiente de la tecnología y con un nivel de abstracción lo más alto posible.
- Técnicas de descripción formal: Usar lenguajes de modelamiento formales, pero que al mismo tiempo sean claros para todo el equipo de trabajo, lo cual implica usar en algunos casos notaciones semiformales que ayuden a la comprensión de las descripciones.
- Abstracción del sistema: Aspectos fundamentales para la abstracción correcta de un sistema como: Descripción del propósito y funcionalidades, entorno del sistema, asociaciones del sistema, definir patrones o comportamientos similares.

Los aspectos fundamentales son reflejo de un orden metodológico para el desarrollo de una solución, en el presente trabajo se cubren todas las fases del M.C.S donde los productos esenciales más importantes siguen el orden de la tabla 2 adaptada de [15].

Modelo para la construcción de soluciones	**Fases Mecanismo de Integración de Pandaboard en el IDE de Arduino en un Contexto de IoT**
Fase 1 (“Estudio de Prefactibilidad”)	Formulación de la Propuesta
Revisión y compromiso 1 (Transición a Fase 2)	Aprobación de la Propuesta y nombramiento del Director del Trabajo
Fase 2 (“Formulación del Proyecto”)	Elaboración del Anteproyecto respectivo
Revisión y compromiso 2 (Transición a Fase 3)	Capítulo 1 Monografía Capítulo 2 Monografía Capítulo 3 Monografía
Fase 3 (“Ejecución del Proyecto”)	Capítulo 4 Monografía Capítulo 5 Monografía Capítulo 6 Monografía
Revisión y compromiso 3 (Transición a Fase 4)	
Fase 4 (“Validación de la Solución”)	Capítulo 7 Monografía Sustentación Trabajo de Grado

Tabla 2. Fases M.C.S en el Desarrollo del Sistema

2.3 Internet de las cosas (IoT).

En su definición y representado por su nombre, el IoT son las cosas cotidianas que se conectan a Internet, aunque en realidad es mucho más que eso, formalmente, se puede decir, que el Internet de las cosas se trata de una red que interconecta objetos físicos valiéndose del Internet, estos objetos se valen de sistemas embebidos, o también llamado, hardware especializado, que permite no solo una conectividad a Internet, sino que además permiten programar eventos específicos en función de las tareas que le sean dictadas remotamente. La idea general es hacer un poco más interactivos todos los objetos de uso cotidiano,

ideas como las de un hogar inteligente (domótica) o Smart House, son ya una realidad [16] [17].

Después de la aparición de la red de redes (Internet) y con ella de la Web, además del auge del Internet móvil, estamos inmersos o ad portas de una tercera, y potencialmente más disruptiva fase: el llamado “Internet de las Cosas” (Internet of Things, IoT) [18]; el IoT hace referencia a un mundo conectado hasta el último extremo, donde objetos y seres físicos interaccionan con entornos virtuales de datos en el mismo espacio y tiempo, soñamos con poder medir y controlar por completo nuestro entorno [18]. Es una revolución tecnológica que representa el futuro de la computación y las telecomunicaciones, y su desarrollo depende de la innovación técnica en un número importante de campos, desde redes de sensores hasta nanotecnología [19], sin embargo con el creciente uso de tecnologías portables y de las mayores facilidades de alcance para una persona promedio, el Internet de las Cosas (IoT) aún se encuentra con varias barreras que pueden representar un retraso para su desarrollo, las tres barreras principales son la implementación de IPV6, la energía para alimentar los sensores y el acuerdo sobre las normas [20]. En contraste con esto y en efecto con el beneficio del procesamiento de información integrada, los objetos industriales y los cotidianos tomarán características y capacidades inteligentes, además, podrán tomar identidades electrónicas que serán accedidas remotamente, o ser equipados con sensores para detectar cambios físicos alrededor de ellos; eventualmente, incluso partículas tan pequeñas como un grano de arena podrán ser etiquetadas y conectadas a la red, tales desarrollos convertirán los objetos meramente estáticos de hoy en nuevos objetos dinámicos, con inteligencia embebida en nuestro entorno, y estimulando la creación de una serie de innovadores productos y servicios [19].

Aún con esto, el Internet de las Cosas ha seguido avanzando; por ejemplo: ya ha logrado que la Internet sea sensorial (temperatura, presión, vibración, luz, humedad, estrés), lo que nos permite ser más proactivos y menos reactivo, IoT

representa la próxima evolución de Internet dado que los seres humanos avanzan y evolucionan mediante la conversión de datos en información, conocimiento e inteligencia, IoT posee el potencial para cambiar el mundo tal como lo conocemos [20]

En la figura 1 se observa el diagrama en bloques de la Topología de Comunicaciones de la Internet de las Cosas. En el primer bloque (Ambiente Arduino) es donde se encuentra todo lo relacionado con los sensores de Arduino, el ambiente de programación de Arduino y el IDE de Arduino. En el segundo bloque (Pandaboard) se encuentra la puerta de enlace (Gateways) encargada de la comunicación entre los bloques. En el bloque final (Nube) es donde se alojan todos los datos obtenidos del primer bloque y de la implementación que se les hizo en el segundo bloque.

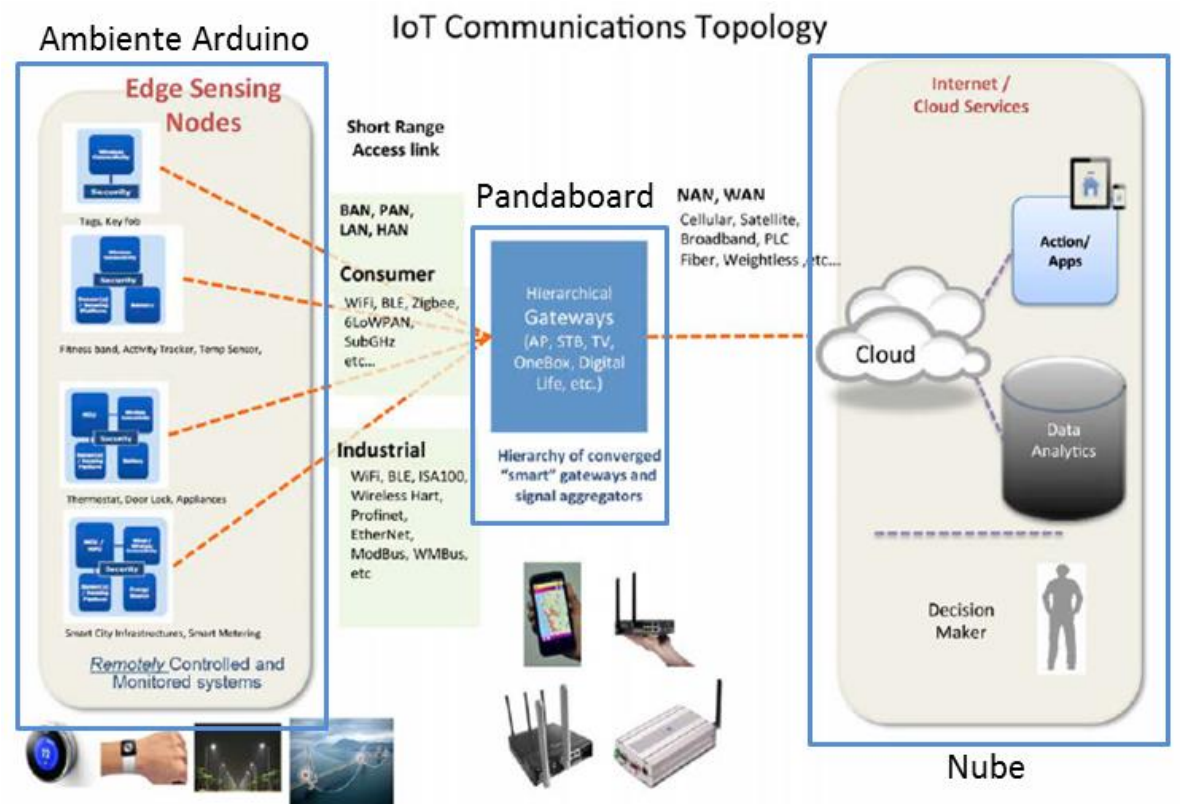

Figura 1. Diagramas en Bloques IoT [Amtel IoT, 2014].

Para nuestro estudio se divide la infraestructura Internet de las Cosas (IoT) según se muestra en la figura 2. Se divide en dos bloques: Plataformas de Hardware y Ambientes de Desarrollo Software. En el primer bloque (Plataformas de Hardware) se hace una descripción de los módulos que lo componen: Arduino, Pandaboard, Redes de Sensores (Wireless Sensor Network, WSN) y Gateway. En el segundo bloque (Ambientes de Desarrollo Software) se hace una descripción de los módulos que lo componen: IDE Arduino, Librerías, Script, C++ y Sistema Operativo. Los cuales son descritos en detalle en los numerales 2.4, 2.5 y 2.6.

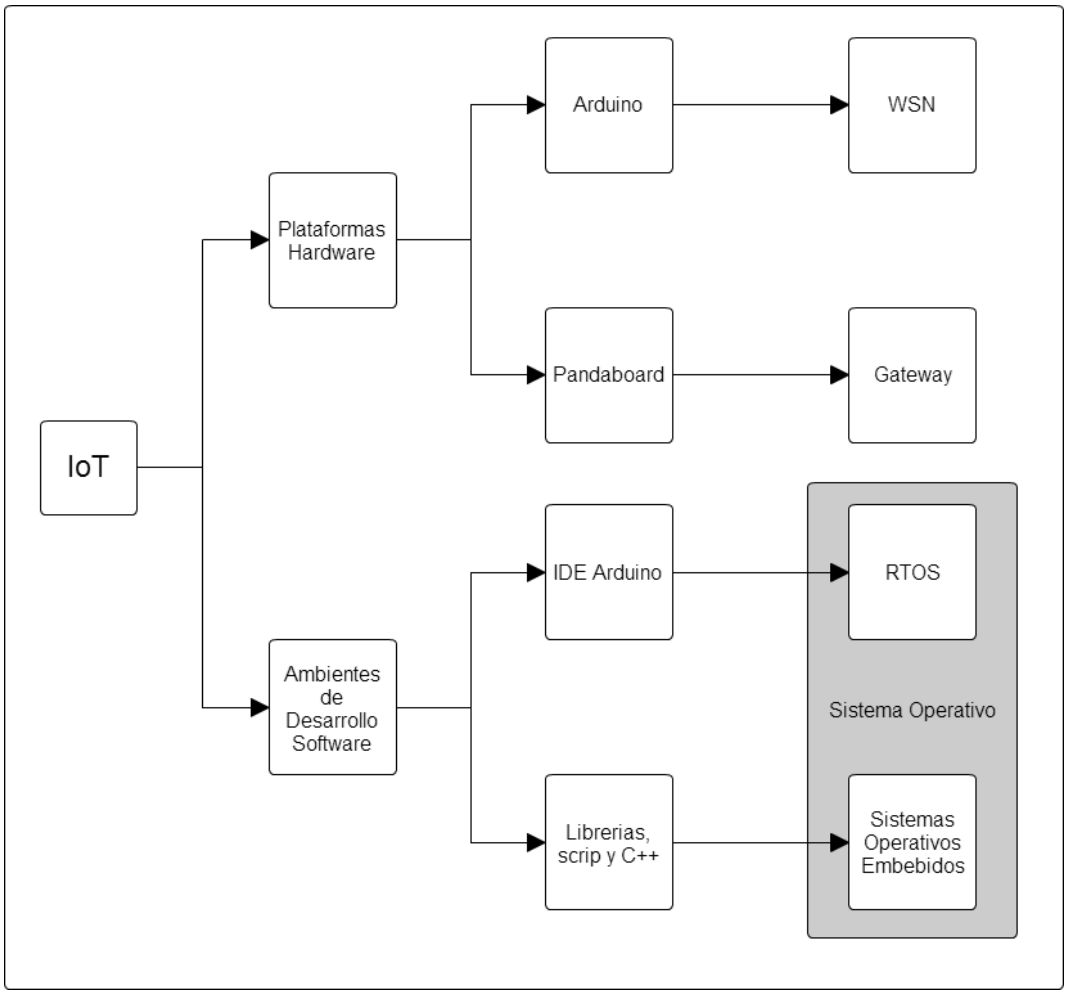

Figura 2. Diagrama de Conceptos Fundamentales

2.4 Plataformas Hardware.

Inicialmente se hace una definición de lo que son las Plataformas de Hardware Libre y las implicaciones que estas tienen dentro del desarrollo del mundo hardware.

Las plataformas de hardware libre o electrónica libre como ya se había escrito en la descripción del contexto general de este trabajo de grado, son aquellos dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita [21]. El hardware libre toma las mismas ideas del software libre para aplicarlas en su campo, en lo referente a las cuatro libertades: libertad de uso, de estudio y modificación, de distribución, y de redistribución de las versiones modificadas, su objetivo es crear diseños de aparatos informáticos de forma abierta, de manera que todas las personas puedan acceder, como mínimo, a los planos de construcción de los dispositivos. [22] El crecimiento de este sector es muy rápido, con unas cifras de beneficios que pueden alcanzar los mil millones de dólares en 2015, una de las iniciativas más conocidas es Arduino, una plataforma de hardware para controlar dispositivos electrónicos, aunque existen también compañías como SparkFun, productora de tarjetas electrónicas, que genera cerca de 10 millones de dólares al año y cuenta con una plantilla de 60 trabajadores. [23] Sin lugar a dudas Arduino ha sido el caso más emblemático en el mundo, Arduino es una plataforma de electrónica abierta para la creación de prototipos, basada en software y hardware flexibles y fáciles de usar. [24]

2.4.1 Arduino.

Es una plataforma de hardware de código abierto y de desarrollo de computación física, la cual se basa en una sencilla placa de circuito impreso que contiene un microcontrolador y un entorno de desarrollo que sirve para crear el software (programa) que se usa en la placa Arduino. [25] Arduino tiene una gran variedad

de usos, entre ellos: puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando multitud de tipo de luces, motores, y otros actuadores físicos, también puede crear objetos interactivos y puede leer datos de una gran variedad de interruptores. [25]

Arduino se creó en un centro académico donde los estudiantes se dedicaban a experimentar con la interacción entre humanos y diferentes dispositivos, muchos de estos dispositivos basados en microcontroladores, esto con el fin de generar espacios únicos, especialmente se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos [26].

Es una placa de hardware libre (open hardware), que incorpora un microcontrolador reprogramable, una serie de entrada y salida que permiten conectar allí de forma muy sencilla y cómoda diferentes sensores y actuadores, además es fácil de usar. El microcontrolador en la placa se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing), cuando se crea un proyecto en Arduino, este puede ser autónomo, es decir, puede ejecutarse sin necesidad de conectar a un computador, si bien tienen la posibilidad de hacerlo y se puede comunicar con diferentes tipos de programas (software) que se ejecuten en el ordenador (ej. Flash, Processing, MaxMSP). [26] [27]

La tarjeta Arduino, contiene para interacción con el usuario: trece (13) pines de entrada/salida digitales, seis (6) entradas analógicas y un (1) puerto serial que permite realizar comunicación con otros periféricos, también cuenta con una conexión USB y un pulsador para resetear la placa si se produce un fallo en los procesos que se estén realizando [28].

Además de simplificar el proceso de trabajar con microcontroladores, la tarjeta Arduino ofrece algunas ventajas respecto a otros sistemas:

- Factible: las placas Arduino son más asequibles y factibles comparada con otras plataformas de microcontroladores.
- Multiplataforma: el software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux.
- Entorno de programación sencillo y directo: el entorno de programación de Arduino es fácil de usar para usuarios con niveles básicos de conocimiento (principiantes) y lo suficientemente flexible para los usuarios avanzados, Arduino está basado en el entorno de programación de Procesing, con el que el usuario aprenderá a programar y se familiarizará con el dominio de desarrollo Arduino.
- Software ampliable y de código abierto: el software de Arduino está publicado bajo una licencia libre, y tiene la ventaja de poder ser ampliado por programadores y desarrolladores con experiencia, el Lenguaje puede ampliarse a través del uso de librerías de C++.
- Hardware ampliable y de código abierto: Arduino está basado en los microcontroladores ATMEGA168, ATMEGA328 Y ATMEGA1280. Los planos de los módulos están publicados bajo licencia Creative Commons, esto permite que diseñadores de circuitos puedan hacer su propia versión del módulo, además de poder ampliarlo u optimizarlo [29].

En la figura 3 se muestra una imagen de la Placa Arduino y de sus principales características.

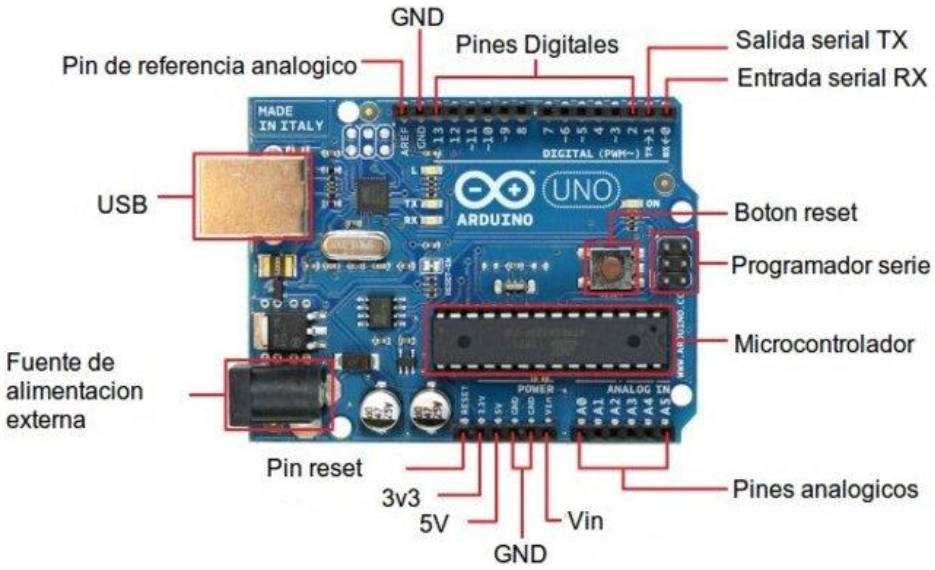

Figura 3. Características de la Placa Arduino UNO

A continuación en la tabla 3 se presentan las características más importantes de la Placa Arduino:

Microcontrolador	ATmega328
Voltaje de operación	5 voltios
Voltaje de entrada recomendado	7 a 12 voltios
Voltaje de entrada limite	6 a 20 voltios
Pines de entrada y salida digitales	14 (de los cuales 6 proporcionan una salida PWM)
Pines de entrada y salida analógicos	6
Corriente de cada pin de entrada y salida	40 miliamperios
Corriente para la salida 3.3 voltios	50 miliamperios
Memoria flash	32KB (ATmega328) de los cuales 0.5KB son usados por el bootloader o gestor de arranque
SRAM	1 KB
EEPROM	1 KB
Velocidad del Reloj	16 MHz

Tabla 3. Características de la Placa Arduino UNO

De entre las placas Arduino se pueden encontrar multitud de modelos, todos especialmente pensados para un fin, compatibles con los shields y módulos oficiales, así como con Arduino IDE [30]. A continuación se enumeran algunas de las principales:

- Arduino/Genuino 101
- Arduino Zero
- Arduino Yun
- Arduino Leonardo
- Arduino Due
- Arduino Mega
- Arduino Ethernet
- Arduino Nano

Todas las placas Arduino están basadas y sus características se desprenden del primer modelo, el Arduino Uno, exceptuando algunas con diseños especiales.

Con esto se concluye la descripción de la placa Arduino y a continuación se presenta la Pandaboard cómo tarjeta de desarrollo del presente proyecto de grado.

2.4.2 Pandaboard.

La Pandaboard es una Single Board Computer (SBC) o mini computador, construido sobre una tarjeta, con microprocesador(es), memoria, entradas y salidas (input/output), de bajo consumo y de baja potencia, usada para desarrollo, la Pandaboard está basada en el Texas Instrument OMAP4460 con su System on Chip (SoC), es ideal para el desarrollo, mejora de plataformas y productos móviles con múltiples funciones, la Pandaboard puede soportar varios sistemas operativos basados en Linux como Android, Chrome y Ubuntu. Es una plataforma diseñada para proporcionar acceso a la mayor cantidad de características del procesador

como sea posible, manteniendo al mismo tiempo un bajo costo, esto permitirá que el usuario desarrolle software para utilizar las características del procesador, además, al proporcionar la capacidad de expansión a través de conectores. [31] Ahora bien, cuenta con una Arquitectura ARM, que es una arquitectura de 32 bits desarrollada para usarse en computadoras personales que manejan un sistema de instrucciones simples, lo que le permite ejecutar tareas con un mínimo consumo de energía y cuenta con una comunidad de desarrollo [32].

La Pandaboard será utilizada como una Puerta de Enlace (Gateway), se hace uso de la tarjeta SD donde se transporta el sistema operativo Linux (Debian), y mediante conexión desde el computador hasta la Pandaboard haciendo uso del cable USB/Serial [31] [32].

En la figura 4 y la tabla 4 se muestran las características de la Pandaboard:

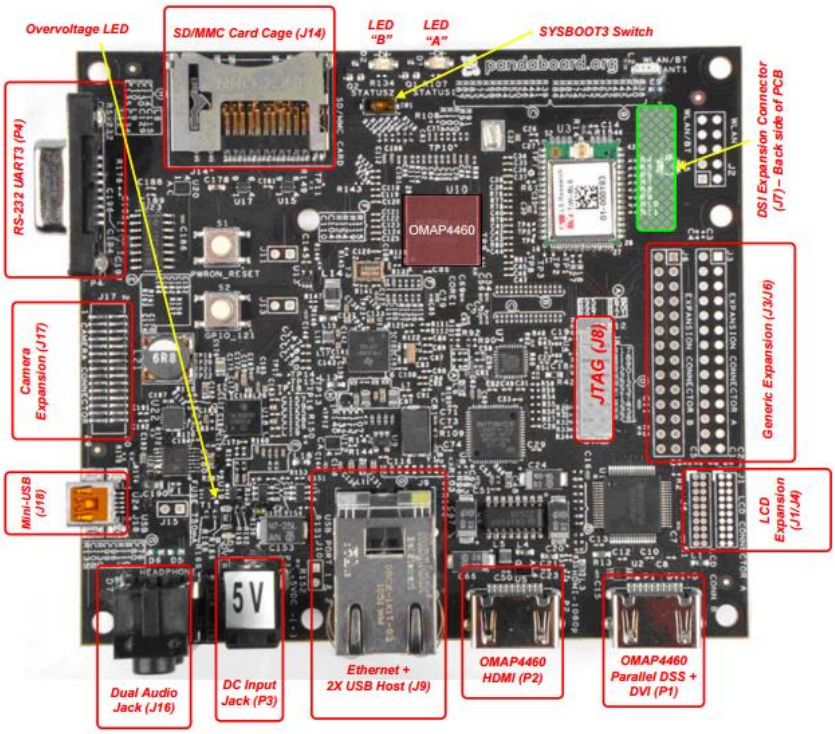

Figura 4. Pandaboard OMAP4460 ES [5]

	Feature	
Processor	OMAP4460	
POP Memory	Elpiga 8Gb LPDDR2 (EDB8064B1PB-8D-F)	
PMIC	TI (TWL6030 Power Management Companion IC)	
Debug Support	14-pin JTAG	GPIO Pins
	UART via DB-9 connector	LEDs
PCB	4.5” x 4.0” (114.3 x 101.6mm)	8 layers
Indicators	3 LEDs (two user-controlled, one overvoltage indicator)	
HS USB 2.0 OTG Port	Mini-AB USB connector, sourced from OMAP USB Transceiver	
HS USB Host Port	Four USB HS Ports, up to 500mA current out on each, two to onboard connectors, two to expansion connectors	
Audio Connectors	3.5mm, L+R out	3.5mm, Stereo In
SD/MMC Conecctor	6 in 1 SD/MMC/SDIO	4/8 bit support, Dual voltaje
User Interface	1-User defined button	Reset Button
	SYSBOOT3 switch	
Video	DVI-D or HDMI	Optional user provided plug-in display
Power Conecctor	USB Power	DC Power
Camera	Not included, but supported via camera expansion connector	
Expansion Connectors (not populated)	See Paragraph 2.17 on page 41	
Parallel LCD Expansion Connectors (not populated)	See Paragraph 2.13.1 on page 32	
DSI LCD Expansion Connector (not populated)	See Paragraph 2.13.2 on page 34	

Tabla 4. Tabla de Características de la Pandaboard

En la figura 5 se muestra la Arquitectura de la Pandaboard:

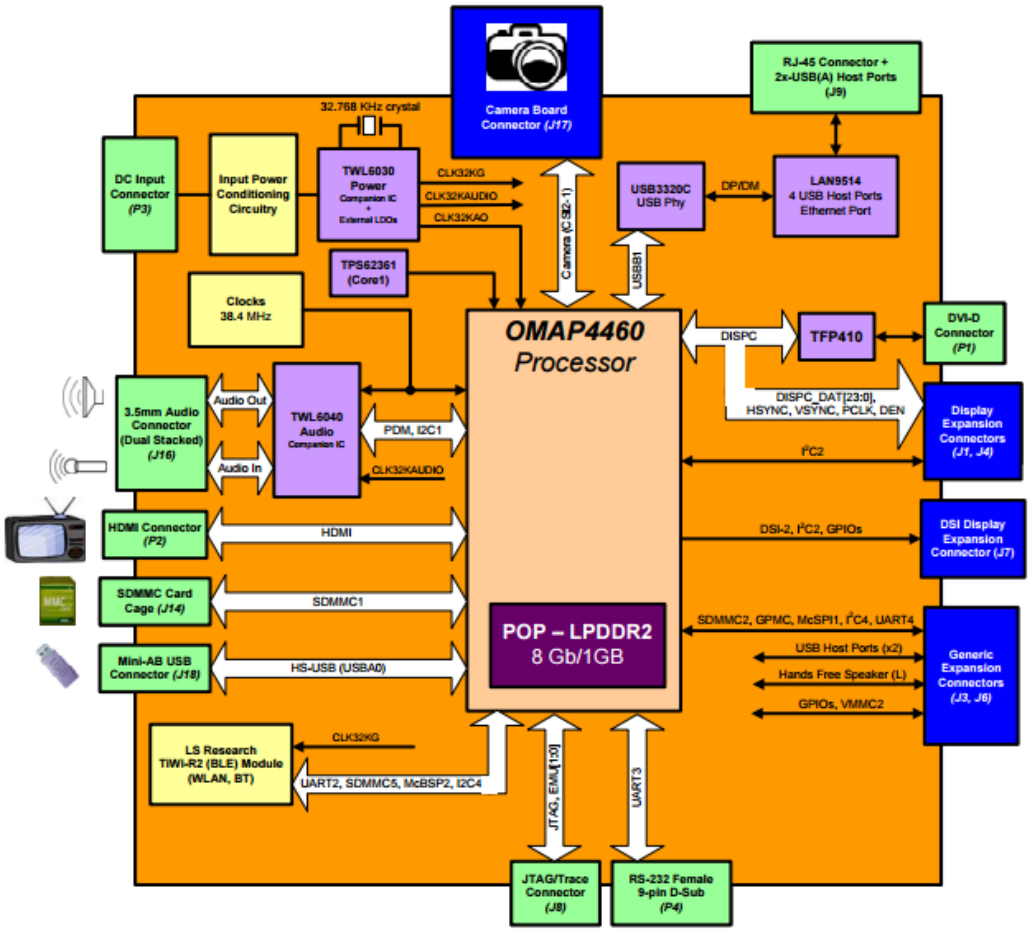

Figura 5. Diagramas en bloque de la arquitectura de la Pandaboard OMAP4460 ES [5]

Con esto se concluye la descripción de la tarjeta Pandaboard. A continuación se hace una definición de las redes de sensores (WSN), que son utilizados como parte del contexto del internet de las cosas (IoT).

2.4.3 WSN.

Las redes de sensores constituyen actualmente una herramienta tecnológica muy valiosa para operar en actividades de campo, recolectar información o incluso

operar equipos a distancia, algunas de las características que deben cumplir los dispositivos que componen redes de sensores (Wireless Sensor Network, o WSN), una WSN está constituida por dispositivos comúnmente denominados “nodos”, los cuales poseen capacidades reducidas de cómputo, sensado y comunicación inalámbrica [33]. Generalmente delegan a otros dispositivos: tareas de gestión de red, enrutamiento de datos y comunicación con pasarelas (gateways) hacia sistemas de computación.

Los nodos que componen las WSN aplicadas a sensado remoto deben cumplir determinadas características para ser confiables y para que su desarrollo sea factible.

Está formada por decenas de nodos, es fundamental el bajo costo de los mismos, de lo contrario, el costo total de la red sería muy elevado y su desarrollo inviable o no práctico.

Los nodos generalmente no tienen fuente de energía recargable, y el mantenimiento por parte del personal capacitado es muy reducido o casi nulo, por lo cual los nodos deben consumir lo mínimo indispensable para su funcionamiento, también es necesario que los protocolos de comunicación presenten especificaciones que permitan maximizar el tiempo de vida de los nodos y de la red, como así también distribuir equitativamente el trabajo entre todos los nodos para balancear el consumo de energía de cada uno, y minimizar el impacto en la red de fallas aleatorias en nodos.

Con esto se concluye la descripción de las redes de sensores (WSN). A continuación se hace una definición de la puerta de enlace (Gateway), que se utiliza como parte del contexto del Internet de las Cosas (IoT).

2.4.4 Gateway

La puerta de enlace (Gateway) es un punto en la red que funciona como entrada hacia otra red, actúa de interfaz de conexión entre aparatos o dispositivos, permite encaminar o enrutar nuestra conexión a Internet, y también posibilita compartir recursos entre dos o más computadora. El propósito fundamental de la Gateway es la de traducir la información del protocolo que se está utilizando en una red inicial, al protocolo usado en la red de destino. Permite interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación, también permite dotar a las máquinas de una red local (LAN) de un acceso hacía una red exterior, generalmente realizando para ello operaciones de traducción de direcciones IP (NAT: *Network Address Translation*). Esta capacidad de traducción de direcciones permite enmascarar las direcciones IP, para dar acceso a Internet a los equipos de una red de área local compartiendo una única conexión a Internet, y por tanto, una única dirección IP externa. [34] [35]

2.5 Otras Plataformas y Dispositivos Hardware.

La brecha digital y quienes se preocupan por acortarla, los altos costos de importación o fabricación de hardware y la comodidad-portabilidad han hecho surgir nuevas tecnologías que se adapten a la demanda de cada cliente, con el análisis hecho a estas nuevas tecnologías, se definieron las siguientes como base para la escogencia de las usadas en el desarrollo del presente trabajo de grado:

- **Plataformas embebidas:** se analizaran los computadores en miniatura basados en el formato de Computadora de Placa Única (Single Board Computer, SBC).
- **Otros Dispositivos Hardware:** se describen otros dispositivos usados en el desarrollo del presente trabajo de grado.

2.5.1 Plataformas embebidas.

También conocidos como Computadores de Placa Única, o Computadoras en un Circuito (Mini Sigle Board Computer), son las encargadas de administrar el resto de subsistemas y de procesar todas las tareas. Entre las plataformas embebidas analizadas, se ha decidido destacar las siguientes:

- Raspberry PI
- Beagleboard
- Intel Galileo Gen 2
- AMD Gizmo Board

Además, se desarrolló una pequeña comparación con las características principales de las mismas.

✓ **Raspberry PI**

La Raspberry PI es un mini computador del tamaño de una tarjeta de crédito, fue desarrollada por la Raspberry Pi Foundation, tiene un chip Broadcom BCM2835 que integra una CPU, GPU, DSP, RAM y controlador USB, también incluye un procesador ARM1176JZ-S a 700 MHz, además, su firmware incluye una serie de modos ”Turbo” para que el usuario puede hacer overclocking (Practica mediante la cual se pretende alcanzar una mayor velocidad de reloj para un componente electrónico) de hasta 1 GHz; este chip también tiene internamente una GPU VideoCore IV a 250 MHz encargada de los gráficos y el inicio, esta GPU soporta OpenGL como API Grafica, dependiendo del modelo que elijamos puede tener una memoria SDRAM de 256 MB o 512 MB de RAM.

La Raspberry PI no incluye disco duro o unidad de estado sólido, sino que utiliza una tarjeta SD para el arranque y almacenamiento de datos a largo plazo, la Raspberry PI funciona fácilmente bajo el sistema operativo Debian y distribuciones

Arch Linux ARM disponibles en su página web para su descarga, y tiene integradas herramientas de desarrollo para utilizar Python como lenguaje de programación principal, por otro lado, dispone de una conexión HDMI para una pantalla externa y un Jack RCA, Jack para audio, ranura para SD y MMC, pines GPIO, conexión de alimentación por microUSB de 5v. El modelo B trae incorporado un conector Ethernet para conectarlo a la red cableada directamente, los dos modelos incluyen pines GPIO para entradas y salidas digitales. [36]

En la figura 6 se muestra la infografía de la Raspberry PI Modelo B:

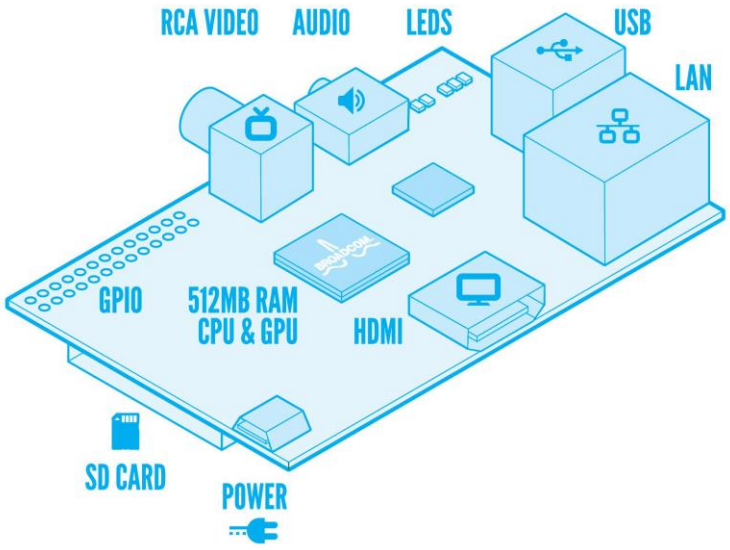

Figura 6. Infografía de una Raspberry PI Modelo B [36]

- ✓ **Beagleboard**

La BeagleBoard es una placa hardware de bajo consumo, producida por Texas Instruments en asociación con Digi-Key. La Beagleboard fue diseñada con el desarrollo de software de código abierto en la mente, y como una manera de probar el procesador de Texas Instruments OMAP3530, tiene toda la funcionalidad de un ordenador muy básico, el procesador OMAP3530 incluye una CPU ARM Cortex-A8, que puede ejecutar Linux, FreeBSD, RISC OS, Android o Symbian,

posee un DSP TMS320C64x acelerador de vídeo y una GPU PowerVR SGX530 para proporcionar acelerado y renderizado de imágenes 2D y 3D compatible con OpenGL. Tiene una única ranura para tarjetas SD y MMC que soporta SDIO, un puerto USB, una conexión serial RS-232, una conexión JTAG, y dos jacks estéreo de 3.5mm para audio.

Incorpora además 256MB de memoria flash NAND y 256 MB de RAM, la placa utiliza hasta 2W de potencia y puede ser alimentado desde el conector USB, o una fuente de alimentación separada de 5V. Debido a su bajo consumo de energía, no requiere refrigeración adicional ni disipadores de calor. [37]

En la figura 7 se muestra el diseño de la Beagleboard:

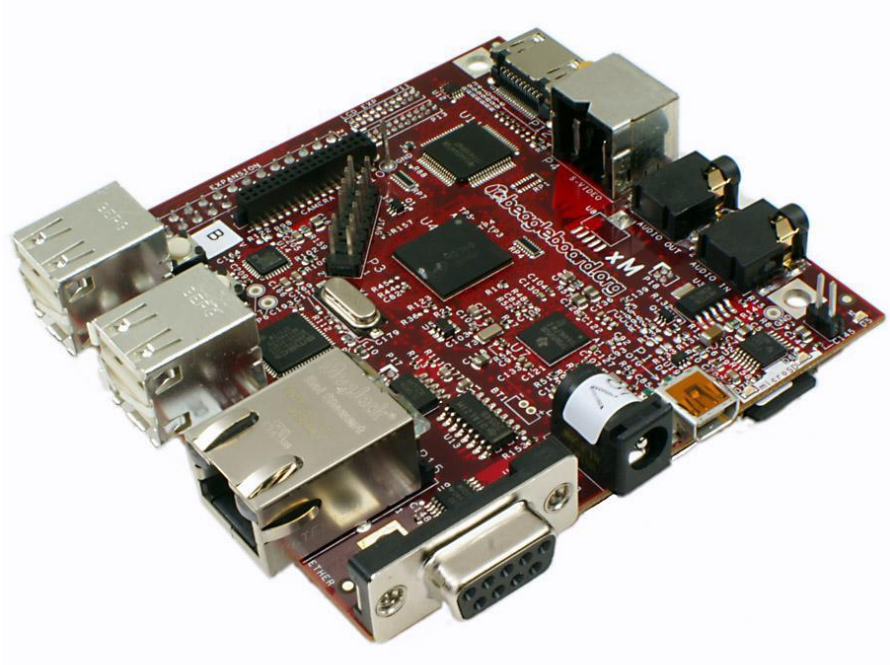

Figura 7. Imagen de la Beagleboard [37]

- ✓ **Intel Galileo Gen 2**

La placa Intel Galileo Gen 2 es la primera de una familia de placas de desarrollo y prototipos certificadas por Arduino basadas en la arquitectura Intel, tiene un procesador Intel Quark SoC X1000, con una arquitectura de conjunto de instrucciones de procesador Intel Pentium de 32 bits, con un solo núcleo y un solo subproceso compatible con ISA, que funciona a velocidades de hasta 400 MHz.

Compatible con una amplia variedad de interfaces de E/S estándar en la industria, entre ellas una ranura mini-PCI Express de tamaño completo, un puerto Ethernet de 100 Mb, una ranura microSD, un host USB y el puerto cliente USB, posee una memoria RAM DDR3 de 256 MB, una SRAM de 512 kb integrada, Flash NOR de 8 MB y EEPROM de 8 kb estándar en la board, más compatibilidad con tarjeta microSD de hasta 32 GB, programable a través del IDE de Arduino, que es compatible con los sistemas operativos Microsoft Windows, Mac OS y Linux, además de ser compatible con la edición Yocto 1.4 Poky Linux, cómo también cuenta con 12 pines GPIO totalmente nativos para mayor velocidad y una resistencia superior de la unidad. El Intel IoT Developer Kit para Intel Galileo Gen 2 incorpora compatibilidad con C, C++, Python y Node.js/Javascript. [38]

En la figura 8 se muestra el diseño de la Intel Galileo Gen 2:

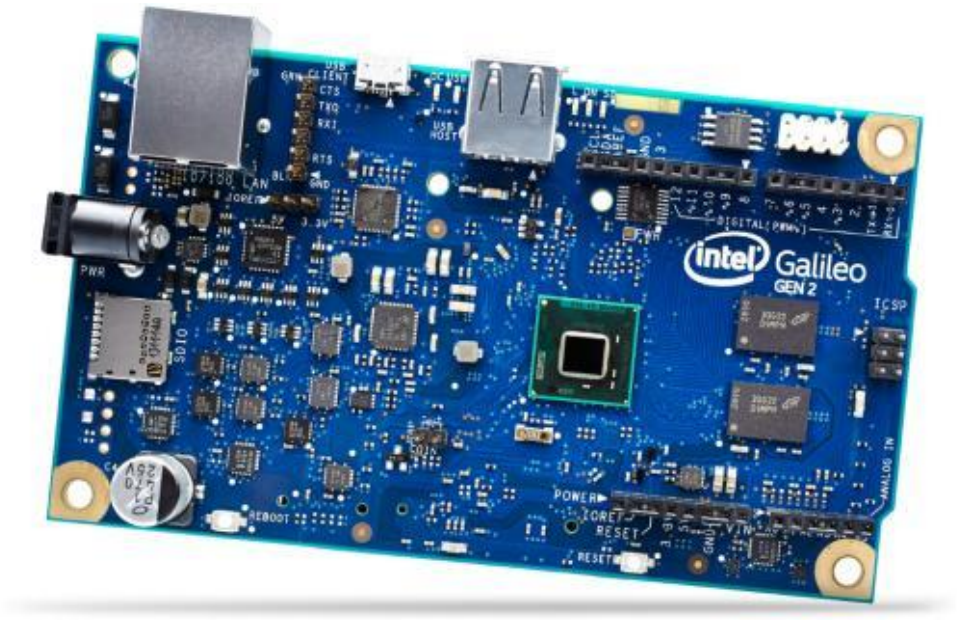

Figura 8. Imagen de la Intel Galileo Gen 2 [38]

- ✓ **AMD Gizmo 2**

La Gizmo 2 es una placa base, de código abierto y es de segunda generación. Ofrece a los programadores un desempeño superior en cómputo y gráficos en una sola plataforma para una amplia variedad de proyectos de desarrollo basados en Linux y Windows.

El chip contiene un procesador SoC AMD Embedded Serie G, una potente y eficiente CPU además de la GPU. La Gizmo 2 ofrece desempeño de cómputo innovador y procesamiento de gráficos excepcional, también promueven la innovación en torno al cómputo heterogéneo de múltiples núcleos, basada en el procesador AMD GX-210HA de doble núcleo, Gizmo 2 es más potente que su predecesora, con una CPU de 1.0 GHz, una GPU de 300 MHz y un TDP de solo 9 vatios en una compacta tarjeta de 4 x 4 pulgadas [39].

La placa Gizmo 2 tiene una entrada de audio/video HDMI, 1 GB de memoria DDR3-1600, una entrada/salida de audio HD, un conector mSATA/mini PCIe, una

ranura para tarjeta microSD, un Gigabit Ethernet en placa, dos USB 3.0, soporte de DirectX 11.1, OpenGL 4.2x y OpenCL 1.2 para procesamiento en paralelo y soporte para memoria Error-Correction Code (ECC) [39].

En la figura 9 se muestra el diseño de la AMD Gizmo 2:

Figura 9. Imagen de la AMD Gizmo 2 [39]

2.5.2 Otros Dispositivos Hardware.

Cable Convertidor USB a Serial RS232: permite la conexión física entre el puerto USB del computador y la Pandaboard, aunque puede verse como obsoleto, resulta muy conveniente usar el protocolo RS232 para el intercambio de información.

En la figura 10 se muestra el cable convertidor de USB a Serial RS232:

Figura 10. Imagen Cable Convertidor de USB a Serial RS232

Memoria microSD: la tarjeta microSD o también llamada *transflash* es básicamente una memoria flash utilizada en equipos portátiles o con requerimientos de memoria externa.

En la figura 11 se muestra la Memoria microSD:

Figura 11. Imagen Memoria microSD

2.6 Ambientes de Desarrollo Software

2.6.1 IDE de Arduino.

Un Entorno de Desarrollo Integrado (IDE), es un entorno de programación o programa informático que ha sido empaquetado como un programa de aplicación, es decir, compuesto por un conjunto de herramientas de programación, más concretamente, el IDE está constituido de: un editor de código de programación, un compilador, un intérprete, un depurador y un constructor de interfaz gráfica (GUI). Un IDE puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios, es decir, los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes [40].

Proporcionan al programador una serie de componentes con la misma interfaz gráfica, con la consiguiente comodidad, aumento de eficiencia y reducción de tiempo de codificación.

Normalmente, un IDE está dedicado a un determinado lenguaje de programación, no obstante, las últimas versiones de los IDEs tienden a ser compatibles con varios lenguajes mediante la instalación de plugins adicionales.

El entorno de desarrollo integrado de Arduino o también conocido como el sistema de desarrollo de Arduino, suministra el entorno que sirve para escribir los programas, compilarlos y posteriormente cargar estos programas en el microprocesador o placa de Arduino mediante el puerto usb, donde harán su proceso de ejecución, esto se hace de una forma sencilla al seguir una serie de pasos, donde primero se ejecuta el (.exe) de Arduino, una vez cargado el constructor de interfaz gráfica (GUI) se configura el puerto a usar y el tipo de placa que se va a utilizar [40].

El IDE de Arduino contiene un editor de texto que es donde se escribe el código, un área de mensajes, una consola de texto, una barra de herramientas para funciones comunes y una serie de menús.

La plataforma Arduino tiene mucha variedad de librerías para interactuar entre el código de la plataforma y el lenguaje seleccionado para programar, la propia plataforma tiene un lenguaje propio de programación, el cual permite aprovechar las características de la placa. El lenguaje usado por Arduino, es una implementación del lenguaje Wiring, y está basado en C/C++. Está licenciado a través de la librería AVR libc [12] y permite el uso de cualquiera de sus funciones. [25]

El lenguaje de programación de Arduino es una implementación de *Wiring* [5], una plataforma de computación física, que a su vez se basa en *Processing*, un entorno de programación multimedia, este lenguaje, conocido como *Wiring*, es un conjunto de funciones que encapsulan el funcionamiento del hardware facilitando el uso del mismo, adicionalmente se puede utilizar las características del lenguaje C++ dentro del entorno de desarrollo lo que permite crear funciones, punteros, clases y objetos e incluso utilizar el lenguaje máquina y otras características propias del compilador para microcontroladores de la familia AVR fabricados por Amtel [6], esto da una gran flexibilidad al momento de crear proyectos complejos y gracias al entorno de desarrollo permite crear aplicaciones rápidamente. También es posible desarrollar librerías que pueden ser instaladas dentro del entorno de desarrollo y existe un gran número de ellas que permiten el manejo de servos, comunicación serial, pantallas LCD, GPS y comunicación WIFI (librería escogida para las pruebas). [25] [40].

2.6.2 Sistema Operativo.

Un sistema operativo es un programa o grupo de programas de proceso, que actúa como interfase entre el programador o usuario y la maquina física (el

hardware), estos programas de proceso, contienen las rutinas de control necesarias para mantener siempre operativos dichos programas [41]. El objetivo básico o primario de un Sistema Operativo es optimizar todos los recursos del sistema para soportar los requerimientos. Dentro de los conceptos o definiciones que puede abarcar un Sistemas Operativo se destacan las siguientes, que hacen parte del desarrollo del presente trabajo de grado:

- ✓ **Sistemas Operativos Embebidos**

Los sistemas operativos embebidos y los de tiempo real son tecnologías inmersas en el diario vivir y generalmente no se identifican fácilmente y tampoco se definen los sistemas operativos que hacen posible su manipulación y que a la vez administran los recursos de dichos dispositivos para maximizar el desempeño [42].

Un sistema operativo embebido es un sistema operativo que se ejecuta sobre un sistema embebido, usualmente tienen algunas características de sistemas de tiempo real, pero a la vez tienen restricciones de tamaño, memoria y energía que los hacen especiales. Este puede ser un sistema de software muy pequeño desarrollado específicamente para ser usado con un algún sistema embebido en particular, o en ocasiones puede ser una versión reducida de algún sistema operativo que se utiliza en una computadora de propósito general. [42][43].

El sistema embebido es un sistema de computación, el cual está diseñado con el propósito de realizar una o muy pocas funciones dedicadas y específicas, empleando para ello una combinación de recursos de hardware y de software [43], se puede decir que muy frecuentemente se habla que es un sistema de computación en tiempo real. Los sistemas embebidos se construyen o diseñan para cubrir una necesidad específica. [44]

Un sistema operativo embebido es aquel que está integrado en los circuitos de los dispositivos electrónicos, entre estos dispositivos se encuentran

electrodomésticos, teléfonos móviles, radios, televisores, automóviles, lectores de códigos de barras, equipos médicos, entre muchos otros.

Estos sistemas suelen tener algunas características de los sistemas de tiempo real pero también tienen limitaciones de tamaño, memoria y consumo de electricidad que los hace especiales y no suelen ser visibles [42].

Algunas características son:

- Fiabilidad y seguridad: Un fallo en un sistema de control puede hacer que el sistema controlado se comporte de forma peligrosa o antieconómica, es importante asegurar que si el sistema de control falla lo haga de forma que el sistema controlado quede en un estado seguro, hay que tener en cuenta los posibles fallos o excepciones en el diseño.
- Eficiencia: Gran parte de los sistemas de control deben responder con gran rapidez a los cambios en el sistema de control.
- Interacción con dispositivos físicos: Los sistemas empotrados interaccionan con su entorno mediante diversos tipos de dispositivos que normalmente no son convencionales (teclados, impresoras): convertidores A/D y D/A, pwm, entradas y salidas digitales paralelo y serie, (interfaces con sensores, actuadores, periféricos especiales). Los componentes del software que controlan el funcionamiento de estos dispositivos son, en general, dependientes del sistema concreto.

- Robustez: Embarcados en sistemas con movimiento o que pueden ser transportados, sujetos a vibraciones e incluso impactos (automóviles, robots, instrumentación portátil). No siempre trabajan en condiciones óptimas de temperatura, humedad y limpieza.

✓ **Sistemas Operativos en Tiempo Real (RTOS)**

Es común entre los ingenieros el uso del término real-time para describir tareas informáticas para las que una respuesta tardía es tan inútil y perjudicial como lo puede ser una respuesta incorrecta, es por esto que se dice, que estas tareas deben tener un tiempo límite (deadline), características básicas que se deben cumplir, debido a que los sistemas operativos embebidos trabajan habitualmente bajo estas condiciones extremas. Es de vital importancia que para el diseño de los sistemas embebidos de tiempo real se conozca a la perfección la capacidad de reacción y comportamiento del hardware y el software que se emplea. Los sistemas de tiempo real (RTOS) deben cumplir tres características básicas:

- Ser determinista: un sistema operativo se dice que es determinista cuando se puede calcular cuál es el máximo tiempo que puede tardar una llamada cualquiera del sistema.
- Garantizar el peor caso de latencia de interrupciones: se llama <<latencia de interrupciones>> al máximo tiempo que transcurre desde que una señal de interrupción llega al procesador hasta que se ejecuta su ISR (Interrupt Service Routine) asociada.
- Garantizar el peor caso de tiempo de cambio de contexto: se llama <<cambio de contexto>> (context switch o thread switch) al proceso de cambiar de una tarea a otra, el contexto de una tarea guarda el estado del procesador justo antes de que otra tarea tome el control de este.

En nuestra propuesta de tesis y para la tarjeta Pandaboard se utiliza Linux Embebido. Linux es un sistema operativo compatible con Unix, desde sus comienzos, Linux se diseñó para que fuera un sistema multitarea y multiusuario, es mucho más seguro que otros sistemas operativos, con características muy peculiares que lo diferencian del resto de sistemas que se pueden encontrar en el mercado, la primera, es que es libre, gran parte de su desarrollo lo realizan voluntarios de forma altruista, la segunda, es que el sistema viene acompañado

del código fuente [45]. El concepto clave de Linux es el kernel, que es el núcleo del sistema, este hace referencia al componente principal de un sistema operativo que es el encargado básicamente de hacer de puente entre las aplicaciones que corren dentro del sistema operativo y del procesamiento de datos llevado a cabo a bajo nivel, es decir, en el hardware [46], además de esto está compuesto por un gran número de programas/bibliotecas que hacen posible la utilización de un núcleo Linux en este caso Linux embebido o empotrado en un sistema embebido como puede ser la Pandaboard (distribuciones hardware afines), teléfonos móviles, robots y otros dispositivos electrónicos. Este núcleo de Linux (embebido) en conjunto con algunas utilidades proporcionadas por versiones de software libre, permiten que se pueda ajustar dentro del limitado espacio de hardware de los sistemas embebidos. Un sistema embebido o empotrado es un sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas frecuentemente en un sistema de computación en tiempo real [47]. Algunas de las características básicas son:

- En el paquete inicial de instalación solo están incluidas las aplicaciones necesarias.
- Es optimizado para permitir utilizar la mínima cantidad de recursos, tanto software como hardware.
- Posee un proceso de arranque rápido.

2.6.3 Librerías.

En el mundo de la informática, una librería es un kit de herramientas software pequeño y autónomo que ofrece una funcionalidad específica al usuario. Se usa para referirse a un programa que contiene varias funciones para lograr un propósito bien definido y específico, estas librerías están diseñadas de tal forma que son fácilmente integradas a otros programas que requieren usar la funcionalidad que la librería ofrece, es posible que una librería utilice otras librerías para completar su funcionalidad [48] [49].

2.6.4 Scripts.

También llamado archivo de órdenes o archivo de procesamiento por lotes, es un archivo que incluye un conjunto de comandos, que se ejecutan desde la primera línea hasta la última línea de forma secuencial, los Scripts son programas, que generalmente son pequeños o simples, y realizan usualmente tareas muy específicas, son un conjunto de instrucciones que por lo regular se almacenan en un archivo de texto plano los cuales deben ser interpretados línea a línea y en tiempo real para su ejecución; este es un factor que los distingue de los programas compilados. Los Scripts pueden ser ejecutados en sistemas de líneas de comando, que son los comandos para UNIX (archivos de extensión .sh) [50].

2.6.5 Lenguaje de Programación C++

Es un lenguaje de programación versátil, potente y general, creado a mediados de 1980 por Bjarme Strousstrup, como extensión del lenguaje de programación C, agregándole mecanismos que permiten la manipulación de objetos. Con base en esas características, y desde el punto de vista de los lenguajes orientados a objetos, se dice que C++ es un lenguaje híbrido, ya que permite programar tanto en estilo procedimental (como si fuese C), en estilo orientado a objetos y también es posible ambas a la vez; Este lenguaje abarca tres paradigmas de la programación: programación estructurada, programación genérica, y programación orientada a objetos. [51]. Su éxito entre los programadores le llevó a ocupar el primer puesto como herramienta de desarrollo de aplicaciones. C++ mantiene las ventajas de C, esto en cuanto se hace referencia a riqueza de operadores y expresión, flexibilidad, concisión y eficiencia, además, C++, eliminó algunas de las dificultades y limitaciones del C original.

Algunas de las principales características de C++, son las facilidades que proporciona para la programación orientada a objetos y para el uso de plantillas o programación genérica (templates), además posee una serie de propiedades

difíciles de encontrar en otros lenguajes de alto nivel. Las principales ventajas que presenta el lenguaje C++ son:

- Difusión: al ser uno de los lenguajes más empleados, posee un alto número de usuarios y existe una documentación muy amplia (libros, cursos, páginas web) dedicados a él.
- Versatilidad: C++ es un lenguaje de propósito general, por lo que se puede emplear para resolver cualquier tipo de problema.
- Portabilidad: el lenguaje está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.
- Eficiencia: C++ es uno de los lenguajes más rápidos en cuanto a ejecución.
- Herramientas: existe una gran cantidad de compiladores, depuradores, librerías.

C++ es considerado por muchos, cómo uno de los lenguajes de programación más potentes, debido a que permite trabajar tanto a alto nivel como a bajo nivel.

2.7 Módulos de Comunicación

WiFi

Es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Estos dispositivos conectados con WiFi, tales como un servidor, una computadora personal, un teléfono inteligente, una impresora, un televisor inteligente, etc., pueden conectarse a Internet a través de un punto de acceso inalámbrico, este punto de acceso tiene por lo general un alcance de unos veinte metros en interiores o espacios cerrados, y una distancia mayor al aire libre o espacios abiertos. “De manera purista vale la pena decir que el acrónimo WiFi se utiliza para identificar los productos que incorporan cualquier variable de la tecnología sin hilos de los estándares IEEE 802.11, que permiten la creación de redes de área

local sin hilos conocidas como WLAN4, y que son plenamente compatibles con los de cualquier otro fabricante que utilice estos estándares.” [52].

Las ventajas que se pueden destacar de las redes wifi son:

- Por ser una red inalámbrica, ofrece una comodidad superior a las redes cableadas, esto se debe a que se puede conectar a la red desde distintos puntos dentro de un espacio lo bastante amplio.

- Una vez haya sido configurada, una red wifi permite el acceso de múltiples computadores, no hay gasto de infraestructura, y se reduce el uso de cables.

- Una de las grandes ventajas, es que la compatibilidad entre dispositivos que usan WiFi es casi total.

ZigBee

Es una tecnología inalámbrica de corto alcance y bajo consumo, es una solución inalámbrica de baja capacidad para aplicaciones en el hogar (seguridad, automatización). ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (wireless personal area network, WPAN). Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías. [53] [54]

En principio, el ámbito donde se prevé que esta tecnología cobre más fuerza es en domótica, automatización industrial, reconocimiento remoto, medicina, etc. La razón de ello son diversas características que lo diferencian de otras tecnologías:

- Su bajo consumo.
- Su topología de red en malla.

- Su fácil integración (se pueden fabricar nodos con muy poca electrónica).

2.8 Computación en la Nube (Cloud computing).

Con los avances en las tecnologías de la información y programación, se han desarrollado nuevas aplicaciones para la Internet, en este caso la computación en la nube o Cloud Computing, llegó para ayudar a reestructurar las infraestructuras de trabajo. Ya se evidencia una mejora en el servicio de tecnologías de la información (TI), las comunicaciones empresariales se han vuelto más unificadas, el lugar de trabajo se vuelve móvil, en el almacenamiento de nuestros datos cada vez utilizamos menos medios físicos (memorias USB, discos duros). Este tipo de avances están permitiendo y ayudando a optimizar los procesos de las empresas, y tiene como objetivo hacer todo más simple, ahorrar en costos de operación y manejar de una mejor manera el presupuesto.

La computación en la nube basa y sustenta su arquitectura en tres pilares fundamentales: software, plataforma e infraestructura, cada pilar cumple con un propósito diferente en la nube y cubre distintas áreas de productos y servicios para empresas y particulares de todo el mundo.

En la figura 12 se muestran las Capas de la Computación en la Nube:

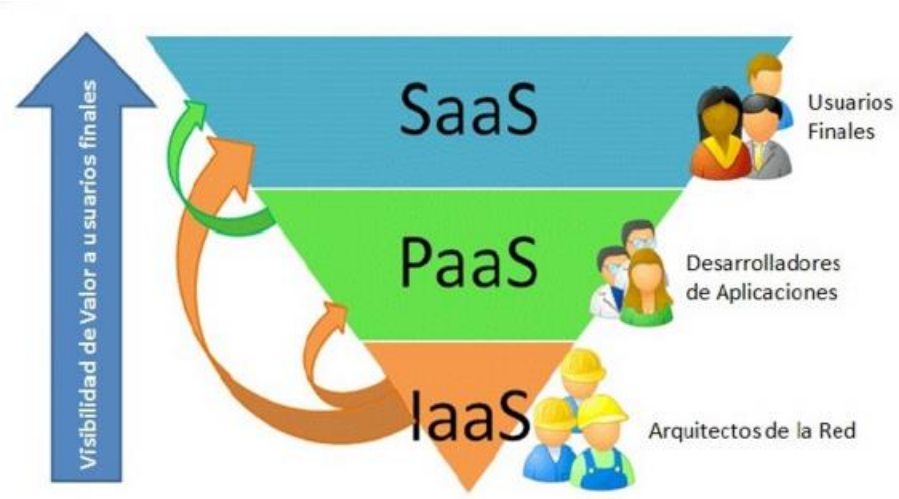

Figura 12. Capas de la Computación en la Nube

- Software como Servicio (SaaS): Se encuentra en la capa más alta de la arquitectura y consiste en un servicio que permite que el consumidor utilice las aplicaciones del proveedor que se ejecutan en una infraestructura de nube. Las aplicaciones son accesibles desde diversos dispositivos cliente a través de una interfaz, como un navegador web, en el Saas el usuario no gestiona ni controla la infraestructura básica de la nube, incluida la red, los servidores, los sistemas operativos, el almacenamiento ni siquiera las capacidades de cada una de las aplicaciones, con la posible excepción de limitados ajustes de configuración de las aplicaciones que son específicos del usuario.

- Plataforma como Servicio (PaaS): Paas es la siguiente capa, básicamente su objetivo se centra en un modelo que proporciona un servicio de plataforma con todo lo necesario para dar soporte al ciclo de planteamiento, desarrollo, puesta en marcha de aplicaciones y servicios web a través de la misma, este servicio permite que el consumidor implemente sobre la infraestructura de nube aplicaciones adquiridas o creadas por el propio consumidor, las cuales se hayan creado usando lenguajes de programación y herramientas compatibles con el proveedor; el usuario no gestiona ni controla la

infraestructura básica de la nube, como la red, los servidores, los sistemas operativos o el almacenamiento, pero tiene control sobre las aplicaciones implementadas y posiblemente sobre las configuraciones del entorno de alojamiento de las aplicaciones.

- Infraestructura como Servicio (IaaS): IaaS es la capa más baja. La idea básica es la de hacer uso externo de servidores para espacio en disco, base de datos, enrutadores, conmutadores (swtiches) así como tiempo de computo evitando de esta manera tener un servidor local, toda la infraestructura necesaria para la conectividad y mantenimiento dentro de una organización. Este servicio permite al consumidor abastecerse de procesamiento, almacenamiento, redes y otros recursos de computación fundamentales, donde el consumidor puede utilizar y ejecutar software arbitrario, puede incluir sistemas operativos y aplicaciones. El usuario no gestiona ni controla la infraestructura básica de la nube, pero tiene control sobre los sistemas operativos, el almacenamiento, las aplicaciones empleadas; y posiblemente un control limitado de determinados componentes que funcionan en red (por ejemplo, los cortafuegos del servidor).

Capítulo 3

3 Alternativas de integración entre Pandaboard y el IDE de Arduino

Con el propósito de seleccionar la integración más adecuada para abordar el desarrollo de los objetivos del presente trabajo de grado, se realiza el siguiente proceso:

- Criterios de selección: Se establecen las características específicas mínimas con las que debe cumplir la forma de integración seleccionada.
- Estudio de las formas de integración disponibles: se realiza una descripción de las formas de integración disponibles, así como de su funcionamiento y arquitectura respectiva.
- Selección de la forma de integración y conclusiones: a partir de los criterios de selección, de los resultados del estudio y de las características de cada forma de integración se realiza una cuantificación ponderada de los resultados con los cuales se selecciona el que registre el valor mayor y que cumpla dichas características.
- Modelado de la Integración Arduino-Pandaboard seleccionada: se hace el proceso de descripción del modelado de la integración.

3.1 Criterios de Selección.

De acuerdo con el propósito que tiene el trabajo de grado y de su carácter académico se plantean los siguientes criterios para escoger la forma de integración que mejor se adapte a las necesidades del presente trabajo de grado, estos criterios de selección son:

- Tiempo de compilación: este criterio es importante, debido a que este proceso requiere de una capacidad amplia en cuanto a procesador y memoria RAM (dependiendo de la complejidad del código y de las dependencias que este requiera).
- Uso de Interfaz gráfica: es importante porque facilita la programación, instalación y corrección de los errores surgidos durante la implementación y desarrollo de la integración. Además que permite al usuario final poder utilizar de una manera más adecuada y amena el entorno de programación.
- Entorno de programación: este criterio es importante debido a que facilita al usuario la programación.
- Facilidad de programación: es importante debido a que permite verificar que tan fácil es para el usuario final la utilización del método de integración y que tanto se facilita la programación con cada uno de ellos.
- Uso del mecanismo de integración: este criterio permite verificar si el uso del mecanismo de integración es el adecuado y por tanto es fácil su manipulación.

3.2 Estudio de las alternativas de integración disponibles.

Teniendo en cuenta los criterios de selección mencionados previamente, se hace una descripción detallada de las alternativas de integración seleccionadas: a) Integración de Pandaboard en el IDE de Arduino, b) Arduino Ejecutándose en Pandaboard y c) La Integración de Pandaboard en el IDE Panda/Arduino, debido a que cumplen con todas las condiciones básicas necesarias para abordar el presente trabajo de grado.

En la sección siguiente se amplía la información relacionada a cada uno de los métodos de integración seleccionados, a través de una definición y descripción de la arquitectura respectiva.

3.2.1 Integración de Pandaboard en el IDE de Arduino.

Mecanismo mediante el cual se hace la integración entre la Pandaboard y el IDE de Arduino. Este mecanismo de integración tiene como propósito la creación de un ejecutable, el cual, en su etapa de desarrollo se basa y hace uso del IDE de Arduino en un computador en el entorno del usuario, y posteriormente, una vez hecha la implementación y creado el ejecutable, se ejecuta en la Pandaboard para verificar el funcionamiento del mecanismo de integración. Haciendo uso de componentes hardware: un computador con buenas características, una Pandaboard, un cable convertidor USB a Serial RS232 y memorias MicroSD. Y software: una distribución de Linux para el computador, un sistema operativo Linux compatible con la Pandaboard.

El mecanismo de Integración de Pandaboard en el IDE de Arduino se compone de los siguientes componentes principales, ver figura 13:

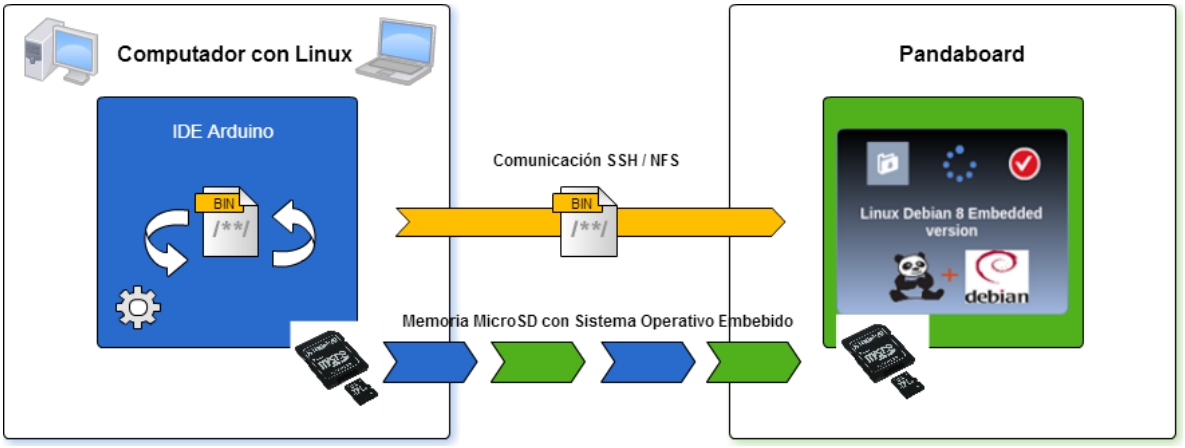

Figura 13. Integración de Pandaboard en el IDE de Arduino

- **Computador (PC):** es uno de los componentes principales, debido a que en él se desarrolla la gran mayoría de los procesos que permiten la implementación del método de integración. Primero se procede con la instalación de un sistema operativo adecuado y que se ajuste a los requerimientos para el desarrollo del

mecanismo de integración, en este caso se instala una distribución de Linux, una vez hecho esto, se procede a la creación del ejecutable (BIN) que posteriormente será enviado a la Pandaboard mediante comunicación SSH o NFS. Desde el computador también se controla y hace la instalación de todos los componentes software que necesite la Pandaboard, este proceso se hace mediante comunicación SSH, NFS o mediante el uso de un cable USB/Serial. También se hace la compilación y programación de las memorias microSD con el sistema operativo adecuado y compatible con la Pandaboard. Al tener un entorno gráfico, todos los procesos de programación, creación y compilación se hacen más rápida y efectivamente, además de contar con unas características de procesamiento mayores (RAM, Procesador).

- **Distribución Linux:** este componente es importante debido a que es el sistema que proporciona todo el soporte software para el desarrollo del mecanismo de integración, que mediante el uso de sistemas de procesamiento cómo C++, y de librerías y scripts, permiten la modificación del IDE de Arduino para la creación del ejecutable (BIN).

- **Memoria MicroSD:** en este componente se instala el sistema operativo embebido que usa la Pandaboard para su funcionamiento.

- **IDE Arduino:** es un componente muy importante, debido a que es el entorno de Desarrollo Integrado (IDE) que sirve para escribir los programas, compilarlos y que posteriormente permite producir el (BIN) para ser ejecutado en la Pandaboard.

- **BIN**: es el componente que lleva toda la información del programa creado para ser ejecutado en la Pandaboard.

- **Comunicación SSH/NFS:** es el componente o vía de comunicación que permite la conexión entre la Pandaboard y el computador, permitiendo la configuración remota de la Pandaboard.
- **Pandaboard:** es el componente principal y es donde se recibe y se hace el proceso de ejecución del programa creado mediante el IDE de Arduino. Aunque los características hardware y de procesamiento no son muy grandes, el hecho de usar un sistema operativo embebido, y de poder ser programada y configurada de forma remota, hace que estas características sean lo suficientemente adecuadas para el proceso de Integración con el IDE de Arduino.

- **Linux Embebido:** proporciona todas las características y componentes software a la Pandaboard. Es un sistema operativo con características básicas, que no hace uso de entornos gráficos y que por tanto elevan el nivel de procesamiento de la Pandaboard al no hacer consumo excesivo de los recursos de esta.

3.2.2 Integración del IDE Arduino Ejecutándose en Pandaboard.

Mecanismo mediante el cual se hace la integración del IDE de Arduino ejecutándose en la Pandaboard. Este mecanismo de integración tiene como propósito lograr ejecutar y correr el IDE de Arduino directamente sobre la Pandaboard, para posteriormente, y una vez hecha la implementación, lograr correr un ejemplo que verifique el funcionamiento. Haciendo uso de componentes Hardware: un computador con buenas características que tiene como único propósito la instalación del sistema operativo embebido en la memoria microSD, una Pandaboard, un cable USB/Serial (solo servirá para la configuración de la Pandaboard) y memorias MicroSD. Y componentes Software: un sistema operativo Linux compatible con la Pandaboard.

El mecanismo de Integración del IDE de Arduino Ejecutándose en Pandaboard se compone de los siguientes componentes principales, ver figura 14:

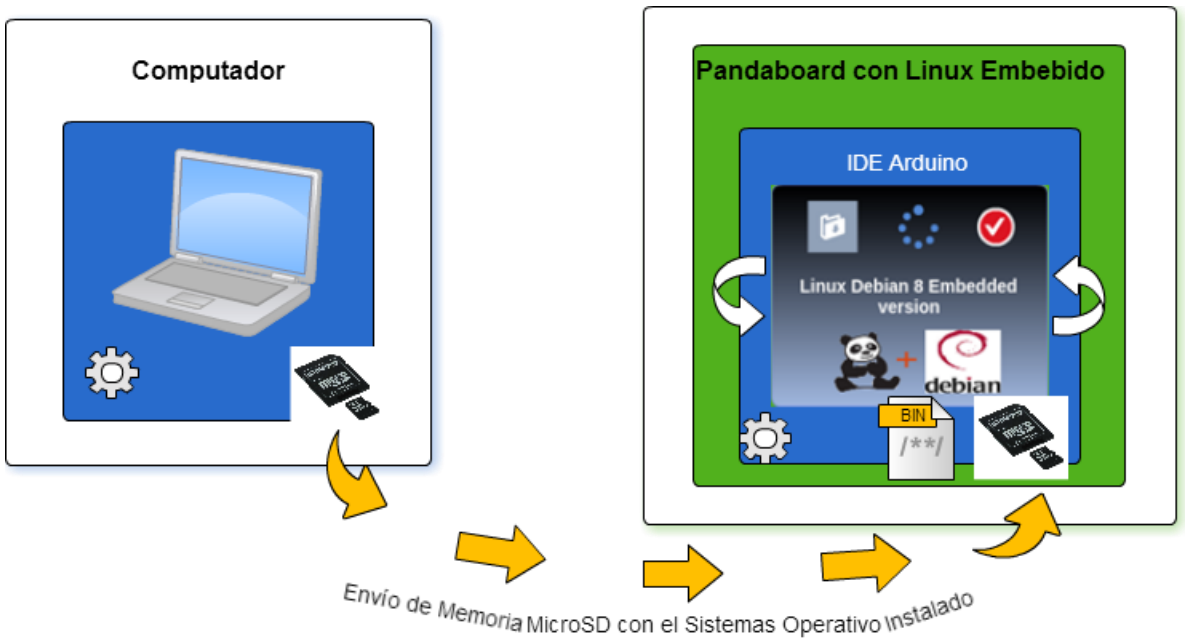

Figura 14. Diagrama del Mecanismo Integración del IDE Arduino Ejecutándose en Pandaboard

- **Computador (PC):** este componente solo es necesario para instalar en la memoria microSD un sistema operativo que sea compatible con la Pandaboard. Además de permitir mediante conexión USB/Serial la configuración de algunos componentes en la Pandaboard. Es necesario especificar que este componente no interviene de forma directa en el funcionamiento del mecanismo de integración.

- **Memoria MicroSD:** en este componente se instala el sistema operativo embebido que usa la Pandaboard para su funcionamiento, este proceso se hace en el computador, para posterior mente, y una vez instalado el sistema operativo, ser integrado a la Pandaboard para su funcionamiento.

- **IDE Arduino:** es un componente muy importante, debido a que es el entorno de Desarrollo Integrado (IDE) que sirve para escribir los programas, compilarlos y que posteriormente permite producir el (.exe) para ser ejecutado

en la Pandaboard. A diferencia del método de integración anterior, en este el IDE de Arduino se instala directamente sobre la Pandaboard.

- **Ejecutable (.bin):** es el componente que lleva toda la información del programa creado para ser ejecutado en la Pandaboard.

- **Pandaboard:** este es el componente principal del mecanismo de integración y es donde se hace el proceso de ejecución del programa creado mediante el IDE de Arduino que corre directamente sobre la placa. Debido a que las características hardware y de procesamiento no son muy grandes en la Pandaboard, y el hecho de usar un sistema operativo embebido, hacen que el proceso de programar, configurar y compilar mediante el IDE de Arduino, sea un proceso largo y dispendioso, pudiendo ocasionar problemas de calentamiento en la placa y demoras en los procesos de compilación del mecanismo de integración.

- **Linux Embedded:** proporciona todas las características y componentes software a la Pandaboard. Es un sistema operativo con características básicas, que no hace uso de entornos gráficos y que por tanto hace la ejecución directa del IDE de Arduino sobre la Pandaboard.

3.2.3 Integración de Pandaboard en el IDE Panda/Arduino.

Mecanismo mediante el cual se hace la integración entre la Pandaboard y el IDE Panda/Arduino. Este mecanismo de integración tiene como propósito la creación de un IDE nuevo, con características semejantes al IDE de Arduino llamado Panda/Arduino, que logre crear un ejecutable para que posteriormente, una vez hecha la implementación, se ejecute en la Pandaboard para verificar el funcionamiento del mecanismo de integración. Haciendo uso de componentes Hardware: un computador con buenas características, una Pandaboard, un cable USB/Serial y memorias MicroSD. Y Componentes Software: una distribución de

Linux para el computador, un sistema operativo Linux compatible con la Pandaboard.

El mecanismo de Integración de Pandaboard en el IDE de Panda/Arduino se compone de los siguientes componentes principales, ver figura 15:

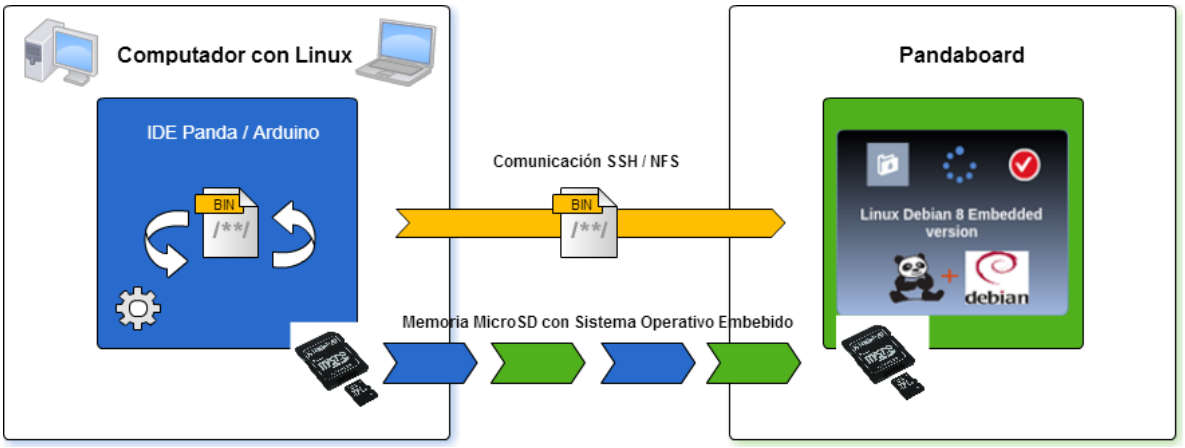

Figura 15. Diagrama del Mecanismo de Integración Pandaboard en el IDE Panda/Arduino

- **Computador (PC):** es uno de los componentes principales, debido a que en él se desarrolla la gran mayoría de los procesos que permiten la implementación del método de integración. Primero se procede con la instalación de un sistema operativo adecuado y que se ajuste a los requerimientos para el desarrollo del mecanismo de integración, en este caso se instala una distribución de Linux, una vez hecho esto, se procede a la creación de un IDE nuevo (IDE Panda/Arduino) para luego crear el ejecutable que posteriormente será enviado a la Pandaboard mediante comunicación SSH o NFS. Desde el computador también se controla y hace la instalación de todos los componentes software que necesite la Pandaboard, este proceso se hace mediante comunicación SSH, NFS o mediante el uso de un cable USB/Serial. También se hace la compilación y programación de las memorias microSD con el sistema operativo adecuado y compatible con la Pandaboard. Al tener un entorno gráfico, todos los procesos de programación, creación y compilación se hacen más rápida y

efectivamente, además de contar con unas características de procesamiento mayores (RAM, Procesador).

- **Distribución Linux:** es un componente muy importante en este método de integración, debido a que además de proporcionar todo el soporte software para el desarrollo del mecanismo de integración, también provee los recursos para la creación del nuevo IDE (IDE Panda/Arduino) mediante el uso de sistemas de procesamiento cómo C++ o Python, de librerías, scripts y de más que sean necesarios, además debe permitir la creación de un ejecutable compatible con la Pandaboard.

- **Memoria MicroSD:** en este componente se instala el sistema operativo embebido que usa la Pandaboard para su funcionamiento.

- **IDE Panda/Arduino:** es el componente principal en este método de integración, debido a que se trata de un nuevo entorno de Desarrollo Integrado (IDE), este proporciona todas las características básicas de un IDE, además que permite producir el ejecutable compatible con la Pandaboard. La principal desventaja de la creación de este nuevo IDE es el proceso de creación del mismo, esto debido a que se hace muy dispendioso y toma mucho tiempo la creación de todas las librerías y entornos necesarios para que el IDE funcione correctamente, yendo en contra del propósito de este trabajo de grado.

- **BIN**: es el componente que lleva toda la información del programa creado para ser ejecutado en la Pandaboard.

- **Comunicación SSH/NFS:** es el componente o vía de comunicación que permite la conexión entre la Pandaboard y el computador, permitiendo la configuración remota de la Pandaboard.

- **Pandaboard:** es el componente principal y es donde se recibe y se hace el proceso de ejecución del programa creado mediante el IDE de Panda/Arduino. Aunque los características hardware y de procesamiento no son muy grandes, el hecho de usar un sistema operativo embebido, y de poder ser programada y configurada de forma remota, hace que estas características sean lo suficientemente adecuadas para el proceso de Integración con el IDE de Arduino.

- **Linux Embebido:** proporciona todas las características y componentes software a la Pandaboard. Es un sistema operativo con características básicas, que no hace uso de entornos gráficos y que por tanto elevan el nivel de procesamiento de la Pandaboard al no hacer consumo excesivo de los recursos de esta.

3.3 Preparación de un modelo básico de evaluación.

El prototipo básico de desarrollo debe cumplir las mismas características y funcionalidades en cada uno de los mecanismos de integración utilizados, con el propósito de garantizar que todos estén bajo las mismas condiciones a la hora de llevar a cabo la selección del mejor.

Para realizar este proceso se tienen en cuenta los siguientes ítems:

- Elementos: los elementos considerados son: computador, Pandaboard, mecanismos de integración (Mecanismo de Integración de Pandaboard en el IDE de Arduino, Mecanismo de Integración del IDE Arduino Ejecutándose en Pandaboard y Mecanismo de Integración de Pandaboard en el IDE Panda/Arduino), ejecutable (BIN)

- Funcionalidad: usando el ejecutable recibir datos de sensores mediante conexión WiFi y enviar estos datos a “un servidor” mediante conexión Ethernet.

- Procedimiento**:** con los dispositivos en funcionamiento construir un ejecutable que sea capaz de recibir datos provenientes de sensores que están integrados a la Pandaboard mediante conexión WiFi y después enviar estos mismos datos a “un servidor” mediante conexión Ethernet.

3.4 Resultados de las evaluaciones.

A continuación se registra los datos de las evaluaciones realizadas, para lo cual son empleadas tablas, y se tienen en cuenta los criterios de selección y un análisis hecho a los prototipos básicos de prueba, además se le da un valor numérico a cada criterio de selección o prueba descrita.

3.4.1 Conceptos generales.

Valor numérico: Este valor numérico va a permitir tener una referencia más exacta de cada mecanismo de integración, permitiendo con esto la escogencia del mismo. Los valores numéricos están entre: uno (1) y diez (10), siendo uno (1) la calificación más baja y diez (10) la calificación más alta.

Abreviaturas: para hacer más fácil el poder plasmar los datos se abreviaron los textos de los criterios y de la evaluación de la siguiente manera. En los criterios de selección: Tiempo de compilación (TC), Uso de Interfaz (UIG), Entorno de Programación (EP), Facilidad de Programación (FP), Uso del Mecanismo de Integración (UMI) y Construcción del Mecanismo de Integración (CMI). En la evaluación: Facilidad de Programación del Ejecutable (FPE), Tiempo de Compilación y Programación (TCP) y Recursos Hardware y Software Requeridos (RHSR)

3.4.2 Resultados de los Criterios de Selección.

En la tabla 5 se registra un análisis de los criterios de selección, y se da una calificación a cada uno de estos, dependiendo de la incidencia que pueda tener respecto al mecanismo de integración. En la figura 16 se muestra la comparación de los Criterios de Selección frente a la calificación dada a cada uno de los criterios de selección.

Criterio	**Mecanismo de Integración de Pandaboard en el IDE de Arduino**	**Mecanismo de Integración del IDE Arduino Ejecutándose en Pandaboard**	**Mecanismo de Integración de Pandaboard en el IDE de Panda/Arduino**
Tiempo de Compilación	Corto, debido al uso de un computador **Calificación: 10**	Largo, debido a que el IDE corre en la Pandaboard **Calificación: 1**	Corto, debido al uso de un computador **Calificación: 10**
Uso de Interfaz Gráfica	La misma interfaz gráfica del IDE de Arduino **Calificación: 10**	No posee interfaz gráfica, debido a las características de la Pandaboard **Calificación: 1**	Una interfaz gráfica similar a la del IDE de Arduino, pero con las características del IDE Panda/Arduino **Calificación: 10**
Facilidad de Programación	Debido al uso del IDE de Arduino, la programación se torna fácil. **Calificación: 10**	La programación es más complicada debido a que no se tiene una interfaz gráfica, ni un entorno de programación. **Calificación: 5**	Debido al uso de un IDE propio y el cual está basado en Arduino, la programación se torna fácil. **Calificación: 10**
Uso del	Su manipulación	Su manipulación	Su manipulación

Mecanismo de Integración	es fácil, debido a que está basado en el IDE de Arduino **Calificación: 10**	se torna algo más complicada, debido al no uso de una interfaz grafica **Calificación: 5**	es fácil, debido a que se tiene un IDE propio además está basado en el IDE de Arduino **Calificación: 10**
Construcción del Mecanismo de Integración	Su construcción es menos complicada debido a que se usan las librerías del IDE de Arduino para su compilación, además solo se modifican las nacerías y que permitan el desarrollo del mecanismo **Calificación: 9**	Su construcción se torna más complicada debido a la complejidad de la instalación del IDE de Arduino directamente sobre la Pandaboard, conllevando con ello problemas de compilación y tiempos de ejecución. **Calificación: 3**	Su construcción es más complicada debido a la dificultad de crear un IDE nuevo y que sea totalmente funcional para el proceso de desarrollo del mecanismo de integración **Calificación: 3**

Tabla 5. Criterios de Selección en Mecanismo de Integración Escogidos

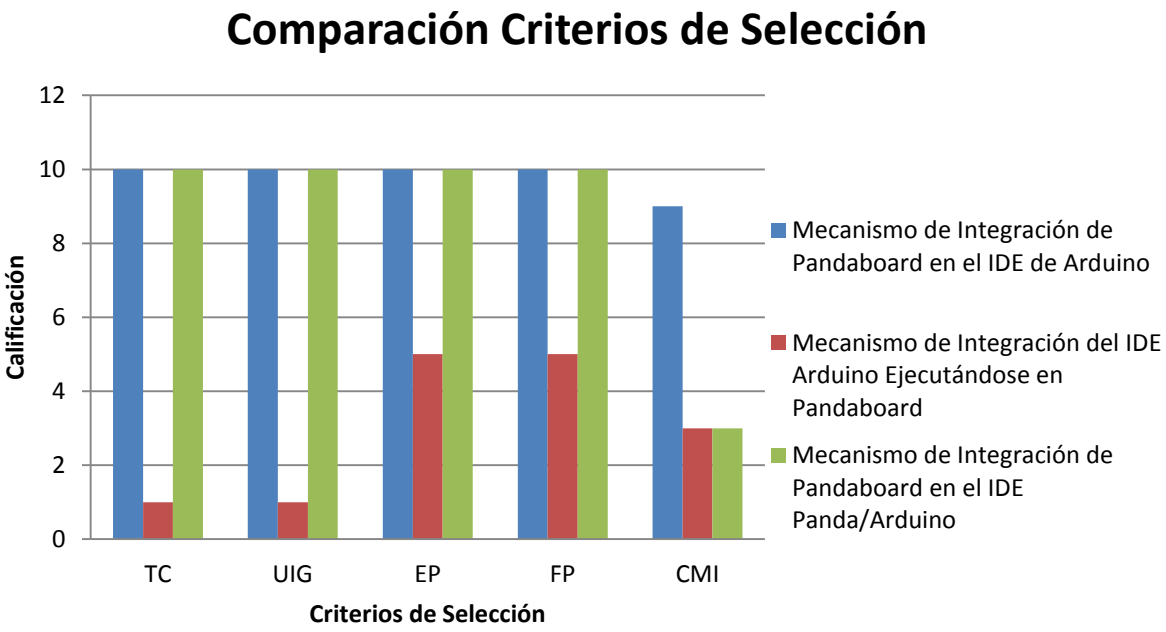

Figura 16. Diagrama de Comparación de Criterios de Selección

El total de calificación de los mecanismos de integración son los siguientes:

Mecanismo de Integración de Pandaboard en el IDE Arduino: 49 puntos.

Mecanismo de Integración del IDE de Arduino Ejecutándose en Pandaboard: 15 puntos.

Mecanismo de Integración de Pandaboard en el IDE Panda/Arduino: 43 puntos.

3.4.3 Conclusiones

- El mecanismo de integración de Pandaboard en el IDE de Arduino es el que muestra más altos porcentajes en la calificación.

- El mecanismo de integración de Pandaboard en el IDE Panda/Arduino, también muestra altos porcentajes en la calificación.

- El mecanismo de integración del IDE Arduino Ejecutándose en Pandaboard es el mecanismo que más bajo porcentaje de Calificaciones muestra

- En la Construcción del Mecanismo de Integración es donde más diferencia se nota entre el mecanismo de integración de Pandaboard en el IDE de Arduino y los restantes mecanismos.

3.5 Selección de la forma de integración.

A partir de los criterios de selección, además de las características de cada forma de integración, así como de la cuantificación ponderada de los resultados, se concluye que el mecanismo de integración más adecuado para el desarrollo del proyecto de grado es el Mecanismo de Integración de Pandaboard en el IDE de Arduino. Esto se evidencia en un mejor uso de los recursos tanto hardware como software, también el uso de una interfaz gráfica y un entorno de programación le dan ventaja y facilitan la programación. Al estar construido su IDE con base en el IDE de Arduino, y al solo modificar una pocas librerías, le permiten tener todas las ventajas que proporciona Arduino para la construcción de un ejecutable funcional. Debido a que la mayoría de procesos de programación y de compilación se hacen desde el computador, esto permite que la Pandaboard tenga que hacer procesos más simples, y por tanto saca ventaja de sus características.

3.6 Modelado del mecanismo de integración entre la Pandaboard y el IDE de Arduino.

3.6.1 Introducción.

En este apartado se analizan los diferentes aspectos relacionados con el proceso de construcción del Mecanismo de Integración de Pandaboard en el IDE de

Arduino. Se inicia con la descripción del negocio, luego se transformaran los requerimientos derivados de las necesidades identificadas en el capítulo 1, en requerimientos funcionales y no funcionales.

3.6.2 MODELADO INICIAL DEL NEGOCIO.

Para este modelo se describen las entidades de negocio. Estas se encuentran compuestas por una serie de procesos y actores relevantes que soportan el sistema original, este proceso es de gran importancia debido a que permite observar el contexto en el cual el sistema será implementado. Como punto de referencia, se declara como sistema inicial, el procedimiento de creación de un programa mediante el uso del IDE de Arduino y la placa de Arduino para su ejecución, y se observan las dificultades que pueda conllevar este proceso.

En el modelado inicial del negocio se identifican los actores iniciales que componen el sistema. Los actores y las características principales:
Programador: es el encargado de la creación de la aplicación (ejecutable), del uso de la interfaz gráfica, de la interacción con el ambiente y entorno de programación, además del correcto manejo de la Pandaboard.

Instalador del Sistema Operativo (S.O.): posee contacto directo con el sistema, se encarga de Gestionar la aplicación y los procedimientos para que el Programador la pueda usar.

Modelo de casos de uso del Negocio.

En la figura 17 se encuentra el diagrama del modelo inicial de negocio.

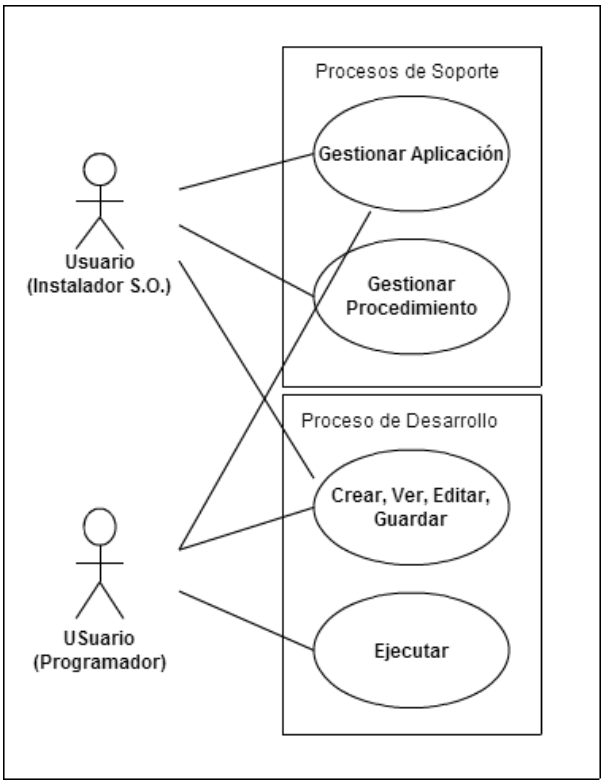

Figura 17. Diagrama de Casos de Uso del Negocio

3.6.3 MODELO DEL AMBIENTE DEL MECANISMO DE INTEGRACIÓN.

El siguiente modelo, describe la captura de requisitos, lista de características y descripción de actores de un Mecanismo de Integración de Pandaboard en el IDE de Arduino, destacando que el presente proceso es fundamental para realizar un aporte al desarrollo de los objetivos propuestos en la sección 1.5.

Con la necesidad de aprovechar todas las características técnicas y funcionales que tiene la Pandaboard, pero además dotando al sistema de un mecanismo que permita una fácil programación e interacción con la misma, todo esto mediante el uso de las funcionalidades y ventajas que nos brinda Arduino con su IDE.

3.6.3.1 Declaración de requisitos

La identificación y definición de los requisitos de un sistema, es el eje fundamental en el desarrollo de una correcta solución usando el M.C.S. Para el presente sistema las necesidades iniciales fueron identificadas en el proceso de búsqueda de un proyecto para el desarrollo del trabajo de grado. Al contar con las tarjetas Pandaboard, se inició el estudio y análisis de las mejores alternativas para iniciar el desarrollo del mismo, en donde se encontró muy poco material y publicaciones, siendo esto un nicho donde desarrollar nuestro proyecto.

3.6.3.2 Lista de características del sistema

Para recolectar la lista de características del sistema se analiza el funcionamiento del mecanismo de integración y se integra con los requerimientos propuestos en 3.6.3.1, la interacción de los usuarios con el mecanismo de integración, se hace a través del cliente. El cliente tiene la posibilidad de usar todas las opciones del mecanismo de integración, que se hace mediante el uso del entorno de programación, y que permite desarrollar la aplicación (ejecutable), que después será enviada a la Pandaboard, en donde se ejecutara y hará el proceso para el cual fue programada. Las características que resumen este flujo de trabajo se encuentran en las tablas 6, 7, 8, 9 y 10.

Nombre	**Programación de la Aplicación**	
Definición	Gestión de la Aplicación (ejecutable), haciendo uso de la interfaz gráfica y del entorno de programación	
Variables	Nivel de prioridad	Crítica
	Riesgo	Crítico, esta característica es base fundamental de las otras, sin la programación de la aplicación no es posible completar el procedimiento.

Tabla 6. Característica: Programación de la Aplicación

Nombre	Compilación de la Aplicación	
Definición	Con la aplicación ya programada y desarrollada, permite la creación del ejecutable (BIN)	
Variables	Nivel de prioridad	Crítica
	Riesgo	Crítico, esta característica es fundamental debido a que la compilación permite producir el ejecutable, para luego ser enviado a la Pandaboard

Tabla 7. Característica: Compilación de la Aplicación

Nombre	Envío de la Aplicación	
Definición	Con la aplicación ya programada, desarrollada y compilada, se debe enviar a la Pandaboard, esto se hace mediante conexión SSH o NFS.	
Variables	Nivel de prioridad	Importante
	Riesgo	Ordinario, depende de que haya conexión para poder funcionar.

Tabla 8. Característica: Envío de la Aplicación

Nombre	Ejecución de la Aplicación	
Definición	Con la aplicación ya lista, se debe ejecutar en la Pandaboard, esto se hace automáticamente.	
Variables	Nivel de prioridad	Importante
	Riesgo	Ordinario, depende de que la aplicación este creada correctamente.

Tabla 9. Característica: Ejecución de la Aplicación

Nombre	**Recolección y envío de datos**	
Definición	Con la aplicación ya lista y ejecutándose en la Pandaboard, esto se hace automáticamente, permite la recolección de los datos provenientes de los sensores, que después se envían a un servidor.	
Variables	Nivel de prioridad	Importante
	Riesgo	Ordinario, depende de que haya conexión para poder funcionar.

Tabla 10. Característica: Recolección y Envío de Datos

4 Implementación del mecanismo de integración entre la PandaBoard y el IDE de Arduino

En la figura 18 se muestra el diagrama del mecanismo de integración con la Pandaboard seleccionado para ser implementado:

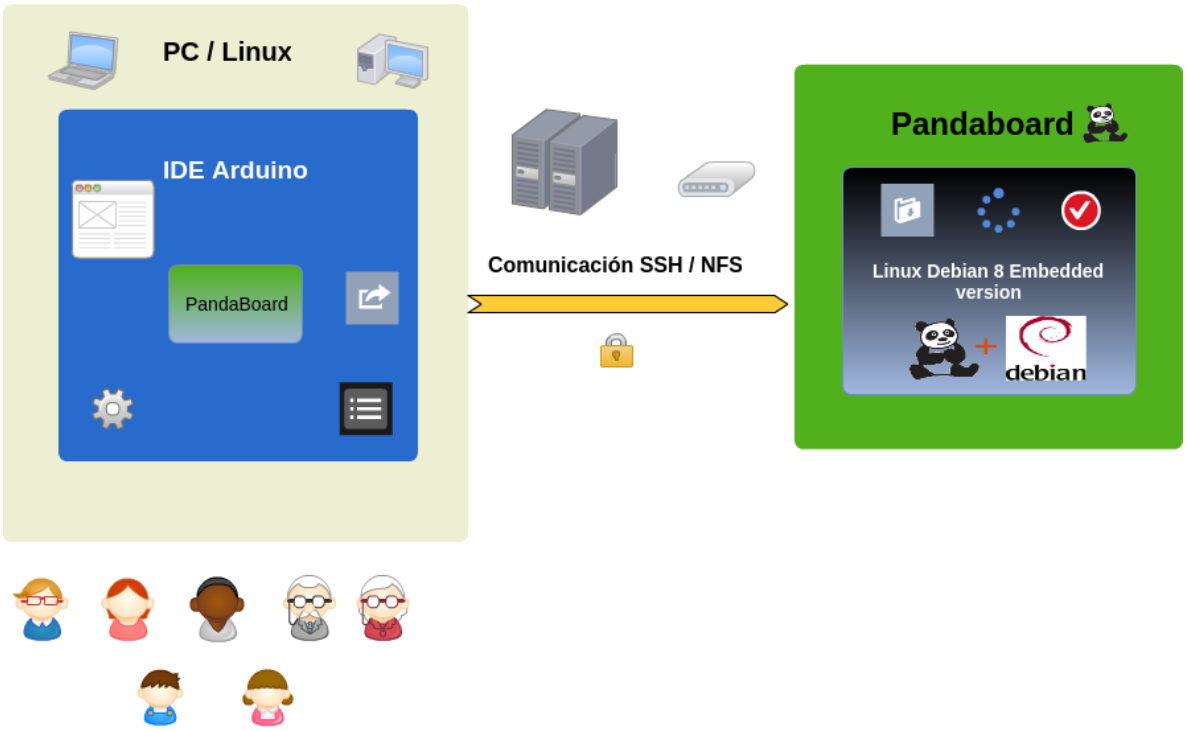

Figura 18. Diagrama de la Implementación del Mecanismo de Integración con la Pandaboard

4.1 Sistemas Operativos Embebidos.

Antes de iniciar las pruebas con los sistemas operativos embebidos se debe verificar el funcionamiento de la Pandaboard con un entorno de validación [55]. Una vez comprobado el funcionamiento, se busca una distribución embebida que sea compatible con la arquitectura de la tarjeta, para montarla en una memoria

SD. En el proceso de selección se contemplaron las distribuciones Ångström (Yocto), Linaro, Debian 8 y Ubuntu 14, todas ellas en sus versiones para la arquitectura armv7. [56]

Las siglas SD_1, SD_2 y SD_3 se utilizarán para relacionar las tres tarjetas de memoria disponibles; sobre estas se montan los distintos sistemas operativos para las distintas pruebas y configuraciones.

4.1.1 Descripción, prueba y selección del sistema operativo embebido.

Para seleccionar el sistema operativo embebido se realizaron algunas pruebas que se describen a continuación:

- Ångström es una distribución que está basada completamente en paquetes y está orientada a sistemas embebidos, sin embargo, la interacción en la etapa inicial es difícil. Se obtiene a través de una “imagen” [57] que contiene una serie de paquetes fusionados dentro de un archivo o sistema de archivos de esta, la cual provee unas funcionalidades básicas de Ångström [57]. En general, hay poca satisfacción en el uso del sistema operativo, dada la mala calidad en la construcción de este y de los productos software construidos para el mismo. En este caso, se descubrió que eran muchos los programas básicos que no se encontraban y que tienen otras distribuciones de Linux, como software de edición (gedit o nano), gestores de red (NetworkManager), gestores de instalación (apt get), etc. La idea es contar con un sistema operativo relacionado con este estudio y no la construcción de un sistema operativo desde el inicio, ya que ese no es el objetivo de este trabajo de investigación.

- La segunda opción que se probó fue Linaro en su versión de Android Kitkat liberado para Pandaboard que no incluye los compiladores de GCC y G++. Según experiencia de usuarios, el sistema se bloquea al abrir un navegador, la

conexión vía Wifi es inestable y la configuración para Ethernet estaba desaparecida. A pesar de ello se utilizó en este proyecto instalándolo en una de las tarjetas SD [58]. El principal problema encontrado se dio al inicio, pues un mensaje indicaba que se debía reiniciar el sistema, y aunque repetidamente se realizó esa acción, el sistema jamás arrancó.

- Otros sistemas utilizados son Ubuntu y Debian en sus versiones 14 y 8 respectivamente. El primero cuenta con características similares a su versión de escritorio y aunque en principio esta versión estaba dirigida hacia la parte multimedia de dispositivos móviles con ARM, ahora está enfocada sobre servidores ARM [59]. El primer sistema requería de la intervención del usuario para continuar su ejecución hasta la sección de login y password, razón por la cual fue descartado. El segundo, es un sistema que provee un entorno sencillo para prototipado rápido y desarrollo de aplicaciones, cuenta con un buen número de paquetes para este propósito y se compila nativamente por defecto. Se encuentra disponible para ARM (armel, armhf, arm64), MIPS y arquitecturas PowerPC. En el proceso de instalación de estos sistemas [60] se encontraron pequeñas diferencias entre ellos pero para los dos es necesario asegurar cual es la versión del kernel para las secciones finales del proceso.

En una etapa posterior es utilizada la versión 15.01 de Linaro (Ubuntu 14.10) [61], la cual cuenta con soporte para WiFi con conexión, pero no es configurable para efectos de uso como punto de acceso para internet; la dirección MAC cambia aleatoriamente después de cada reinicio [62] y la comunicación a través de ssh no es posible debido a que no está configurado apropiadamente. Sin embargo la ejecución de los binarios generados a través de la compilación cruzada en este estudio es exitosa; además se corrobora el conflicto con la versión de glibc en Debian 8 que impedía la ejecución de los mismos.

Finalmente se toma como distribución final la versión de Debian 8, ya que como se mencionó anteriormente cuenta con los elementos software básicos y con los

necesarios para este estudio desde los repositorios oficiales. Pero el punto concluyente es que el inconveniente de compatibilidad de los ejecutables con la versión de glibc de este sistema fue resuelto.

A continuación se muestra una serie de figuras en las que se puede observar el despliegue del sistema seleccionado (Debian 8) desde el momento de la conexión vía puerto serial hasta la parte del ingreso del login y password:

En la figura 19 se encuentra ttyUSB0 dentro del directorio *dev*, lo cual indica la correcta conexión a través del cable usb-serial con la tarjeta.

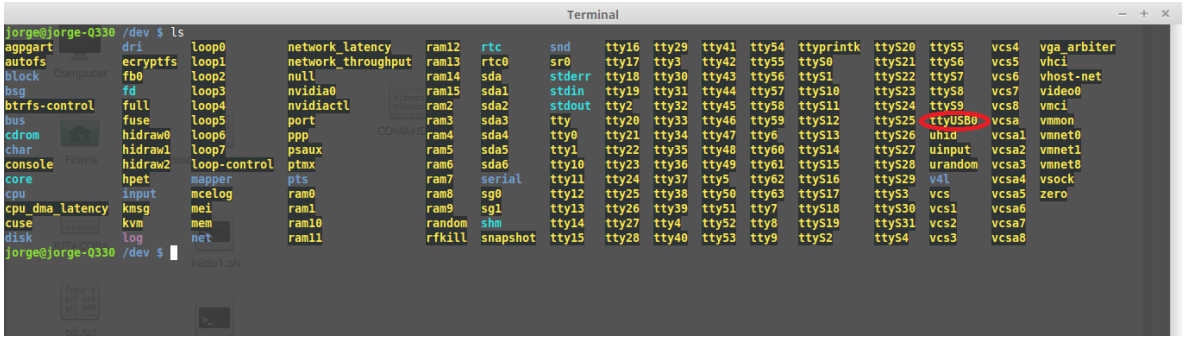

Figura 19. Visualización ttyUSB0 en directorio dev

Para hacer uso de ttyUSB0 a través de la consola serial se debe dar permisos de ejecución. Para el tema de conexión se ha usado una de las interfaces seriales más populares conocida como *cu* [63], la cual hace parte del paquete UUCP y cuya estructura básica para uso es "cu -l [device] -s [speed]" [64], como se ilustra en las figuras 20, 21 y 22.

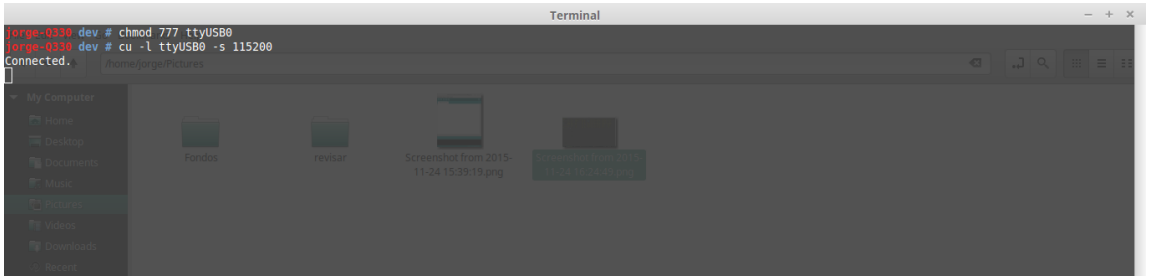

Figura 20. Uso de Consola Serial ‘cu’

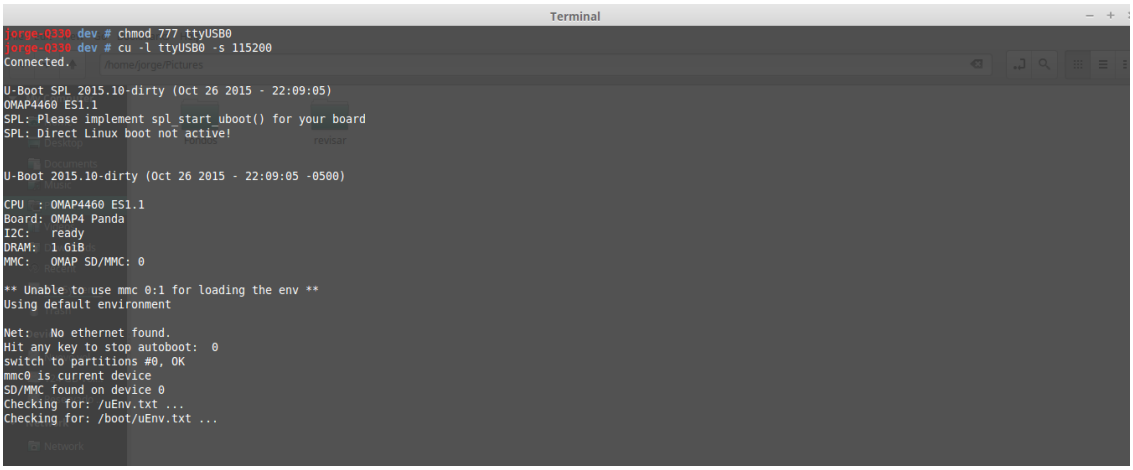

Figura 21. Inicio del sistema debían 8

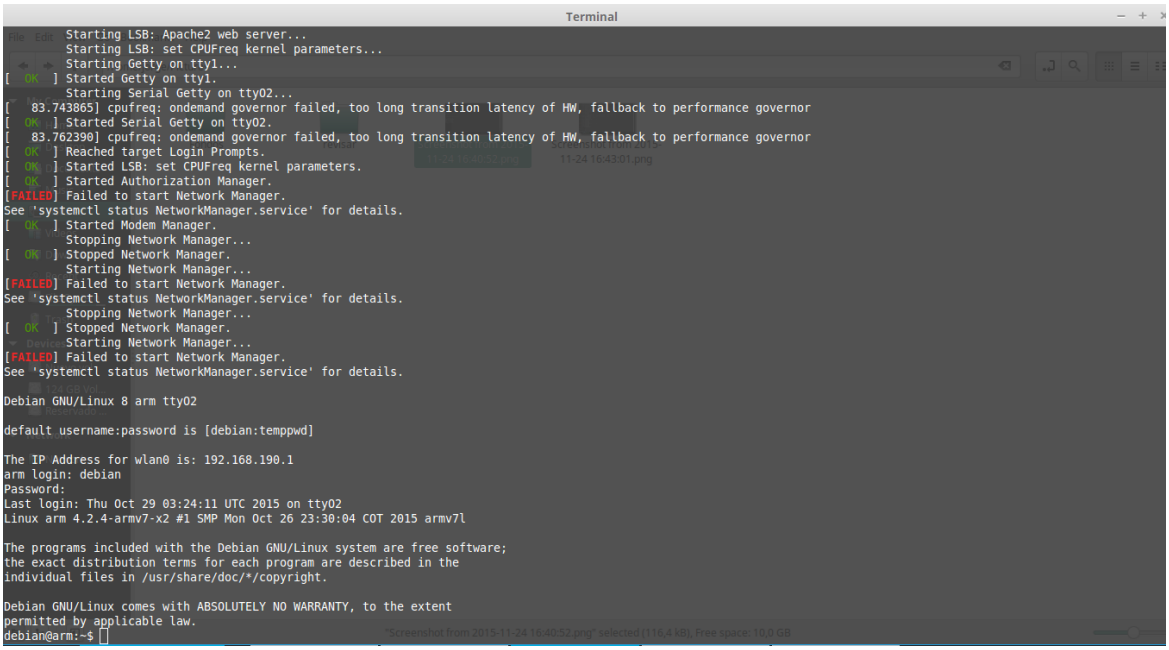

Figura 22. Login y Password sistema embebido

La figura 23 muestra la disponibilidad de los programas necesarios para la comunicación y el enlace de los binarios con la tarjeta a través de *ssh* y la versión de glibc compatible con los ejecutables productos de la compilación para esta arquitectura.

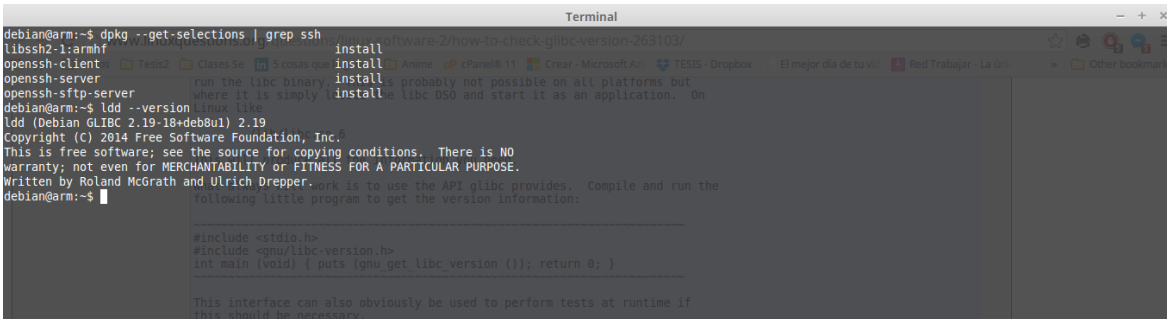

Figura 23. Paquetes de ssh y librería C

4.1.2 Prueba sección WiFi – Access Point.

En esta sección se muestra la implementación de distintos métodos para configurar la Pandaboard como un punto de acceso o Access Point (AP), es decir, como un punto de conexión en redes de área local a fin de brindar interconexión a dispositivos relativamente cercanos siempre y cuando estén configurados y tengan los permisos necesarios [65]. Ya que no existe un procedimiento específico para realizar dicha configuración en esta tarjeta, se prueban acciones definidas para tarjetas similares como Raspberry Pi y BeagleBone.

Estos métodos se desarrollan sobre los sistemas Linaro y Debian, pero para los dos, es necesario garantizar la conectividad vía WiFi y Ethernet. Estas conexiones además deben funcionar implícitamente, sin intervención del usuario. La conexión Ethernet es necesaria a fin de que la Pandaboard pueda recibir la conexión por esta vía y la conexión WiFi para que pueda radiar a dispositivos cercanos cuando esta funcione como Access Point.

El punto de acceso o Access Point debe hacer uso de un “demonio” (daemon) o servicio como airbase-ng; sin embargo, este no tiene soporte para delegar el proceso de autenticación; esto debido principalmente, a que no está pensado para que funcione como un AP con prestaciones tan elaboradas como crear un AP con WPA2. Es por esto que se utiliza hostapd, el cual además de crear puntos de acceso, también se puede usar para servidores de autenticación. Aplica IEEE 802.11 gestión del punto de acceso, IEEE 802.1X / WPA / WPA2 / EAP Autentificadores, RADIUS cliente, servidor de EAP, y servidor de autenticación RADIUS. La versión actual soporta Linux (Host AP, MadWiFi, Prism54 y algunos de los drivers que utilizan el núcleo del subsistema mac80211), FreeBSD (net80211), y DragonFlyBSD. Como se ha dicho, hostapd es un “demonio” por lo que está diseñado para ejecutarse en segundo plano, es decir que su ejecución no puede ser vista por el usuario directamente [66].

En Linaro se puede garantizar la conectividad WiFi en el arranque pero a modo de cliente pues cuando se utiliza como punto de acceso no es posible la conexión. Esto es debido a que como prueba básica no se puede reiniciar el daemon de red en ninguna de las SD en las que se instaló (SD_2 y SD_3). Posteriormente se puede realizar con el uso del comando ‘sudo ifdown wlan0 && sudo ifup wlan0’ aunque algunas veces, al ejecutar el sistema, dejaba de responder. Estos problemas se vieron reflejados en el transcurso de la configuración con la guía que se encuentra en el anexo A presentando inconvenientes con el servicio hostapd. Al final de la misma se indican dos comandos con los cuales se comprueba si el proceso es exitoso o no ('iw wlan0 info' y 'iwconfig') pero el resultado no fue el esperado al menos para el segundo comando. Es por ello que se realizan pruebas para la configuración WL12xx NLCP Build Instructions [67] con el fin de descartar una incorrecta configuración de este driver.

Para la guía [68], el problema está relacionado con una incompatibilidad entre el paquete iptables y la versión Ubuntu 14.10 (Linaro) lo cual impide la generación de

direcciones ip por parte de la Pandaboard hacia los dispositivos que se quieran conectar a ella.

En un tercer método, utilizando el software AP-Hotspot [69] tampoco se pudo realizar la configuración debido a que este programa está desarrollado para solo dos tipos de drivers, que son diferentes al que dispone la tarjeta.

En Debian la conexión por Ethernet y por WiFi se da sin problemas, verificada con un ping hacia cualquier url o host como súper usuario. Continuando con la guía que se encuentra en el anexo A se asigna como ssid “LinaroAP” y como password “linaropanda”. Al finalizar el resultado de los comandos de verificación son exitosos.

Al final debe considerarse que cada vez que se enciende o se reinicia el sistema se debe reiniciar la red, el daemon hostapd y se debe verificar que dentro del Debian de la tarjeta se pueda hacer ping, de no hacerlo, el o los dispositivos podrán acceder a la red pero con servicio limitado (en el caso de que lo hagan) con la excepción de que ningún dispositivo con sistema operativo Windows pudo conectarse.

4.2 Descripción de Utilidades

En esta sección se describirán los elementos necesarios para realizar la compilación cruzada desde un pc hacia la Pandaboard, más específicamente a la arquitectura con la que esta cuenta (armv7).

4.2.1 Descripción porting Arduino.

Básicamente existen dos formas dependiendo de la versión del IDE de Arduino que se esté manejando. Para la versión estándar (1.0.5), la que se instala desde los repositorios propios de Linux (Linux-Mint en este caso) se debe modificar una

serie de archivos (board.txt, platform.txt) y replicar algunos directorios existentes (core, variants) con el fin de partir de la misma estructura y así empezar a adaptarlos de acuerdo a lo que se necesite o a la tarjeta a la que esté dirigida. Cuando aparecieron las tarjetas Galileo y Edison, Arduino tenía un IDE especial para cada una de ellas, pero luego sacó al mercado una versión “genérica” a la que se le podía agregar los componentes necesarios para manejar estas tarjetas a modo de paquetes como es el caso de la versión 1.6.5 sobre la que se trabajó en este estudio.

4.2.2 Herramientas de Desarrollo Pandaboard

Compilador: es un programa informático que traduce un código fuente desarrollado en un lenguaje de programación de alto nivel a un lenguaje de máquina, el cual se considera de bajo nivel, de tal manera que sea entendible y fácil de procesar en el dispositivo que se esté ejecutando [70].

Enlazador: Es un módulo o programa que toma los objetos generados en los primeros pasos del proceso de compilación, la información de todos los recursos necesarios (biblioteca), quita aquellos recursos que no necesita, y enlaza el código objeto con su(s) biblioteca(s) con lo que finalmente produce un fichero ejecutable o una biblioteca[71].

Como ya se mencionó, es necesaria una serie de utilidades para poder generar y principalmente, compilar y enlazar códigos para la arquitectura de la Pandaboard; de tal manera que puedan ejecutarse sobre la misma. Desde el pc se utiliza editores de texto como sublime para desarrollar el código y para las otras dos tareas se instala el compilador cruzado para la arquitectura armv7 de la pandaboard tanto para C como para C++ (gcc-arm-none-eabi). Sin embargo esta herramienta presenta problemas dado que el ejecutable generado a través de ella no trabaja sobre la tarjeta. Es por eso que se utiliza gcc-arm-linux-gnueabihf y

g++-arm-linux-gnueabihf como compiladores para C y C++ respectivamente y el paquete libc6-dev-armhf-cross.

Como parte de este proceso se generan unos ejecutables con extensión elf, bin y con la salida por defecto usando unos ejemplos sencillos (Hola Mundo) y una vez en la tarjeta funcionan correctamente.

Luego de un arduo proceso que se explicará en los capítulos siguientes, se compila un ejemplo haciendo uso de las librerías de Arduino. Inicialmente este programa no compila debido a una incompatibilidad con la librería glibc mencionado en el capítulo de selección del sistema operativo embebido.

A continuación se muestran los makefiles utilizados en el proceso de compilación. El primero “m2”, es utilizado para generar los archivos con extensión .o (objetos) de los archivos con extensión .c y .cpp, los cuales se utilizaran con el siguiente makefile. El segundo “makefile” utiliza los archivos .o del proceso anterior y los reúne y utiliza para generar el ejecutable final, figuras 24 y 25:

main.cpp | m2 | makefile

```
IDIR = /home/jorge/Downloads/arduino-1.5.3-Intel.1.0.4/hardware/arduino/panda
IDIR2 =/home/jorge/Downloads/arduino-1.5.3-Intel.1.0.4/hardware/tools/sysroots/i586-poky-linux-uclibc/usr/include/sys
IDIR3 =/home/jorge/Downloads/arduino-1.5.3-Intel.1.0.4/hardware/tools/sysroots/i586-poky-linux-uclibc/usr/include/linux/
        serial.h
IDIR4 =/home/jorge/Downloads/arduino-1.5.3-Intel.1.0.4/hardware/arduino/panda/cores/arduino
IDIR5 =/home/jorge/Downloads/arduino-1.5.3-Intel.1.0.4/hardware/tools/sysroots/i586-poky-linux-uclibc/usr/src/debug/
        uclibc/0.9.33+gitAUTOINC+946799cd0ce0c6c803c9cb173a84f4d607bde350-r8.4/git/libpthread/nptl/
CFLAGS=-I$(IDIR)
CC=g++
CC2=arm-linux-gnueabihf-g++
CC3=arm-linux-gnueabihf-gcc
ARCH=i386
ARCH2=armv7


#---------------------------------Para generar .o de archivos .cpp-----------------------------------------------------
ScanLM: main.cpp
            $(CC2) -march=$(ARCH2) -w -x c++ -ffunction-sections -fdata-sections -fno-exceptions -fpermissive -c $(IDIR)/
            libraries/WiFi/WiFi.cpp $(IDIR)/libraries/WiFi/WiFiUdp.cpp main.cpp -Wformat-extra-args $(IDIR)/libraries/SPI/
            SPI.cpp $(CFLAGS)/cores/arduino $(IDIR4)/IPAddress.cpp $(IDIR4)/Print.cpp $(IDIR4)/pulseIn.cpp $(IDIR4)/
            RingBuffer.cpp $(IDIR4)/softwarepwm.cpp $(IDIR4)/Stream.cpp $(IDIR4)/TTYUART.cpp $(IDIR4)/UtilMisc.cpp $(IDIR4)
            /UtilTime.cpp $(IDIR4)/variant.cpp $(IDIR4)/WMath.cpp $(IDIR4)/WString.cpp $(CFLAGS)/libraries/SPI $(CFLAGS)/
            libraries/WiFi -L$(IDIR)/cores -larduino -I. #-I$(IDIR3) $(IDIR)/cores/arduino/trace.c $(IDIR4)/Tone.cpp

#---------------------------------Para generar .o de archivos .c-------------------------------------------------------
            $(CC3) -march=$(ARCH2) -w -c -pthread $(IDIR4)/*.c -L$(IDIR)/cores #-include $(IDIR5)/pthread_create.c -I$(
            IDIR5) -I. -lm #
```

Figura 24. Makefile “m2”

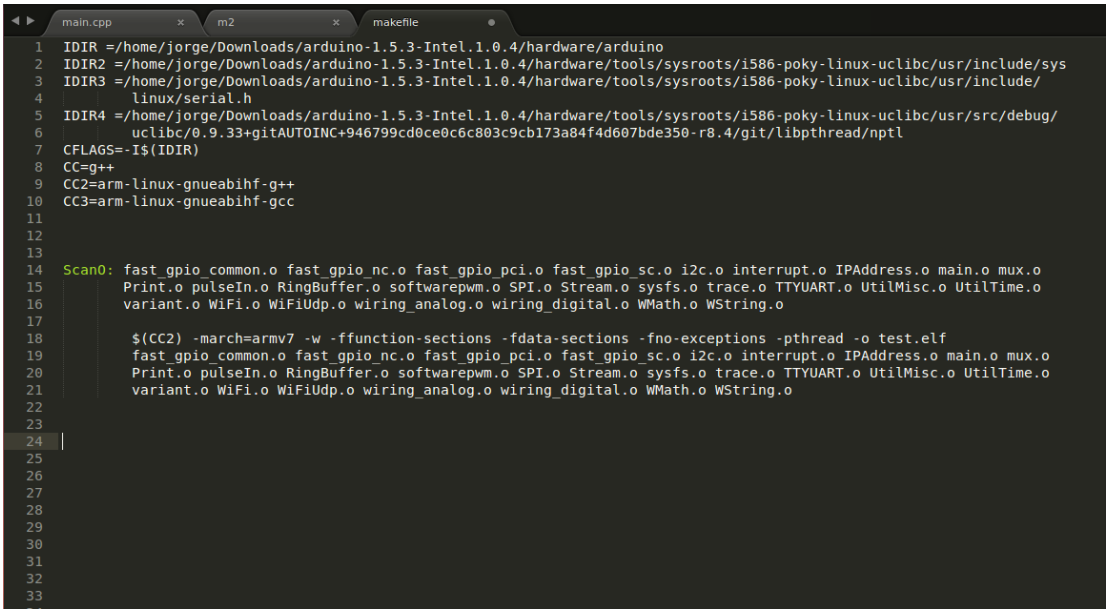

Figura 25. Makefile “makefile”

4.2.3 Herramientas para Automatización.

En esta sección se contempla dentro de la automatización los siguientes ítems: el uso de la tarjeta como Access Point de manera independiente, dado que cada vez que se enciende o reinicia es necesario detener y ejecutar nuevamente los daemons de red y de hostapd; la ejecución de los binarios enviados a la tarjeta sin la intervención del usuario y la compilación de programas a través del uso de archivos makefile.

- Para la primera parte se usa cron, el cual es un administrador regular de procesos en segundo plano que ejecuta procesos o guiones a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Los procesos que deben ejecutarse y el tiempo en que deben hacerlo se especifican en el fichero crontab [72]. La idea es programar el reinicio de los daemon de la red y del hostapd en este archivo. En el proceso se crea un documento con comandos de cron en este fichero, aunque sin poder escribir un contenido en el, por lo

que es difícil corroborar que en efecto la red se está reiniciando, dado que este proceso se ejecuta en segundo plano. Se instala mutt para ver si las tareas programadas en crontab se están ejecutando a través del envío de un correo con el resultado de dicha ejecución, pero no hay éxito en esta tarea. Como alternativa opcional se verifica si el ssid de la Pandaboard aparece en la lista de conexiones disponibles sea desde un smartphone o desde un computador, pero esto no sucede hasta que se reinicia los servicios manualmente.

- En la segunda parte se hace uso de los llamados daemons que son un tipo especial de proceso informático que se ejecuta en segundo plano en lugar de ser controlado directamente por el usuario. Este tipo de programas continúa en el sistema, es decir, que puede ser ejecutado en forma persistente o reiniciado si se intenta matar el proceso dependiendo de configuración del demonio y políticas del sistema [73].

  Para este caso la idea es generar un archivo con una serie de instrucciones en bash e introducirlo en una ruta específica para que se ejecute al arranque. Como primera prueba se sigue la guía [74] pero no pasa nada. Con uno de los scripts hechos para imprimir el resultado de ifconfig se prueba el método [75], pero tampoco hay resultados. Se hace lo mismo en [76], [77] y [78] pero sin nada que indique que está funcionando. De [79] y [80] se puede ver que no basta con copiar el script en cualquier ubicación, hay que considerar el concepto de runlevel, sobre el cual se maneja los tipos de arranque y los privilegios de cada uno dentro de cualquier distribución de Linux, ver figura 26.

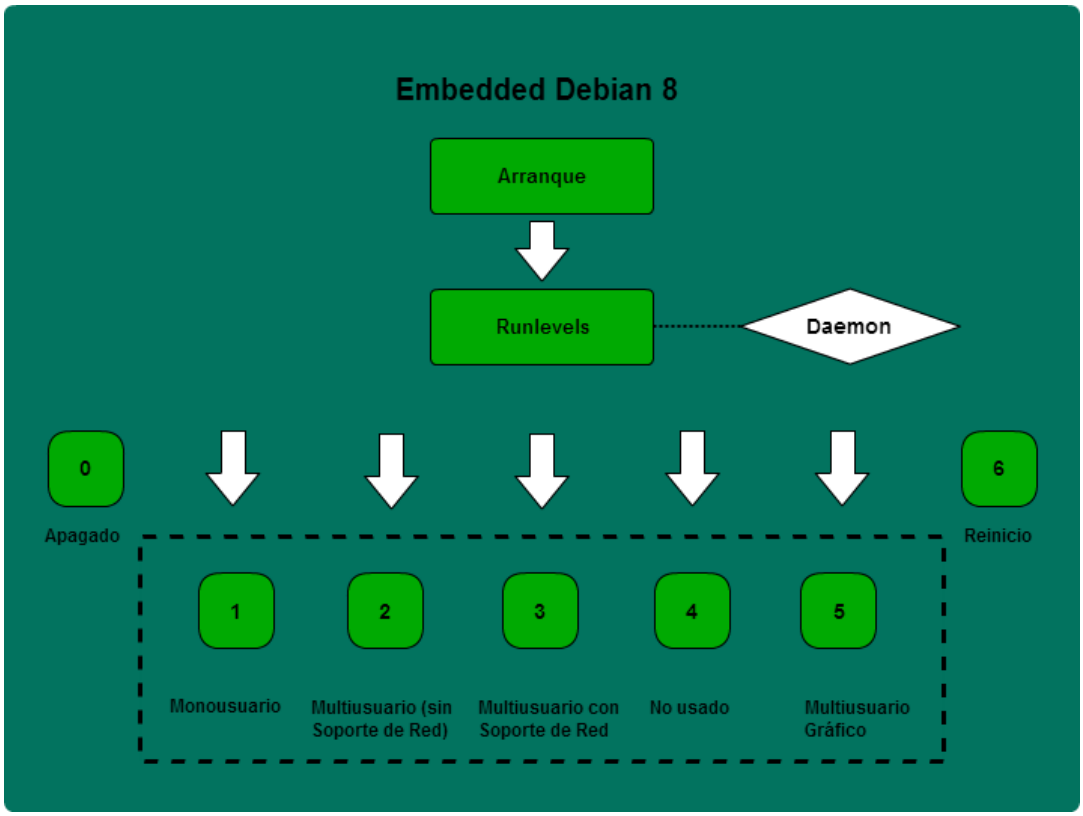

Figura 26. Esquema de Inclusión de daemon y runlevel de sistema

Como un proceso extra se hace uso de distintos software [81] de los que se utiliza solamente los que no tienen interfaz gráfica y pueden ser configurados desde la consola, aunque sin ningún resultado visible tampoco.

- Para la tercera parte, debido a que todas las acciones de compilación se trabajan desde la consola de Linux y es dispendioso ejecutar las instrucciones necesarias cada vez que se desea probar una variación en los códigos y generar sus ejecutables, se hace uso del comando make, como alternativa de automatización. Las instrucciones con los comandos de compilación, los archivos y librerías relacionadas así como sus rutas, son colocadas en un archivo denominado makefile el cual se ejecuta por medio de la instrucción make.

Todas estas pruebas se realizan en la máquina virtual con una distribución Ubuntu de 32 bits dado que se identifica que los errores que aparecen en el host están más relacionados con problemas de la versión de Linux del host que es de 64 bits y no con los códigos, elementos e invocaciones.

4.3 Implementación de Pandaboard en el IDE de Arduino.

4.3.1 Implementación porting IDE Arduino

En principio se trabaja con la versión más reciente de Arduino (1.6.5) la cual se obtiene desde la página principal de Arduino, ya que en el repositorio la versión disponible es la 1.0.5.

Se edita el archivo board.txt en la ruta /home/jorge/arduino1.6.5/hardware/Arduino. Se debe agregar el nombre de la tarjeta a integrar con el IDE, en este caso el nombre es “pandaboard” que efectivamente aparece en la lista de boards dentro del IDE de Arduino.

En este tutorial [82] la tarjeta a relacionar puede hacer uso de las librerías de una tarjeta parecida, la distribución de sus pines es igual así como también el lenguaje de programación, solo se diferencia por la ram. Es por esto que esta guía es limitada para el proceso con la Pandaboard.

En la descripción del porting para las librerías Attiny [83] se puede ver una estructura de ficheros que se prueba pero sin ningún cambio en el IDE.

La característica principal en esta versión de Arduino como se mencionó en un apartado anterior, es la posibilidad de manejar otras tarjetas a través de paquetes que se descargan desde la interfaz de Arduino; los cuales se ubican dentro de una ruta en los ficheros del IDE. Una vez que se instala el IDE de Arduino se crea el directorio oculto .arduino15 en /home/usuario/; inicialmente solo aparecen las tarjetas de AVR, pero una vez que se descarga cualquier paquete desde la

interfaz, dentro de esta ruta se genera el directorio packages con el contenido de la nueva board.

Por seguridad y para posteriores pruebas se crea una copia de la carpeta .arduino15, la cual es nombrada “inicial”

Se hacen una serie de pruebas descargando paquetes desde el IDE y haciendo copias del directorio .arduino15 para cada una de las condiciones, de acuerdo al número de paquetes, y a los paquetes descargados en cada ocasión. La idea de este ejercicio es la de comprobar si la instalación de estos paquetes se puede realizar únicamente desde el IDE o también se puede agregar los directorios con los contenidos necesarios para hacer el porting “externamente”, lo cual sería el ideal a realizar con la Pandaboard.

Tomando como ejemplo la tarjeta AMEL-Tech se puede ver en ella la siguiente estructura de archivos:

- board.txt
- bootloaders
- libraries
- platform.txt
- validation_sme
- variants

Estructura que es bastante similar en comparación a los otros paquetes descargados.

En general se observa que el proceso de instalación y desinstalación es bastante variable, tanto desde el IDE como de manera externa; esto traduce una falta de robustez en el manejo de los paquetes para la versión 1.6.5 de Arduino. Porque mientras para algunos paquetes (como los de Intel), la instalación y desinstalación

era “limpia” y no presentaba problemas con el compilador una vez era reinstalado, para el paquete de AMEL cada vez era mayor el inconveniente y el número de inconvenientes que presentaba a tal punto que aunque se repitieran los procedimientos el error que se presentaba era diferente cada vez. De este experimento se puede concluir que si el IDE de Arduino en su versión 1.6.5 presenta tantos inconvenientes con los paquetes propios, la probabilidad de que funcione con una modificación para soportar la Pandaboard es realmente baja, razón por la cual, esta opción es descartada.

En la máquina virtual con la versión de Arduino 1.0.5 disponible en el repositorio; se siguen los pasos de la guía [83] y trabajan la mayoría de ellos. Se procede a hacer el cambio de la versión anterior de arduino en el host.

Se crea una nueva carpeta llamada 'pandaboard' dentro de /usr/share/arduino/hardware/ en la que se copia el contenido de la carpeta arduino. Se modifica el archivo board.txt de la nueva carpeta y se agrega la línea correspondiente al nombre (Pandaboard ES); se reinicia el IDE y la panda aparece en la lista. Ver figura 27:

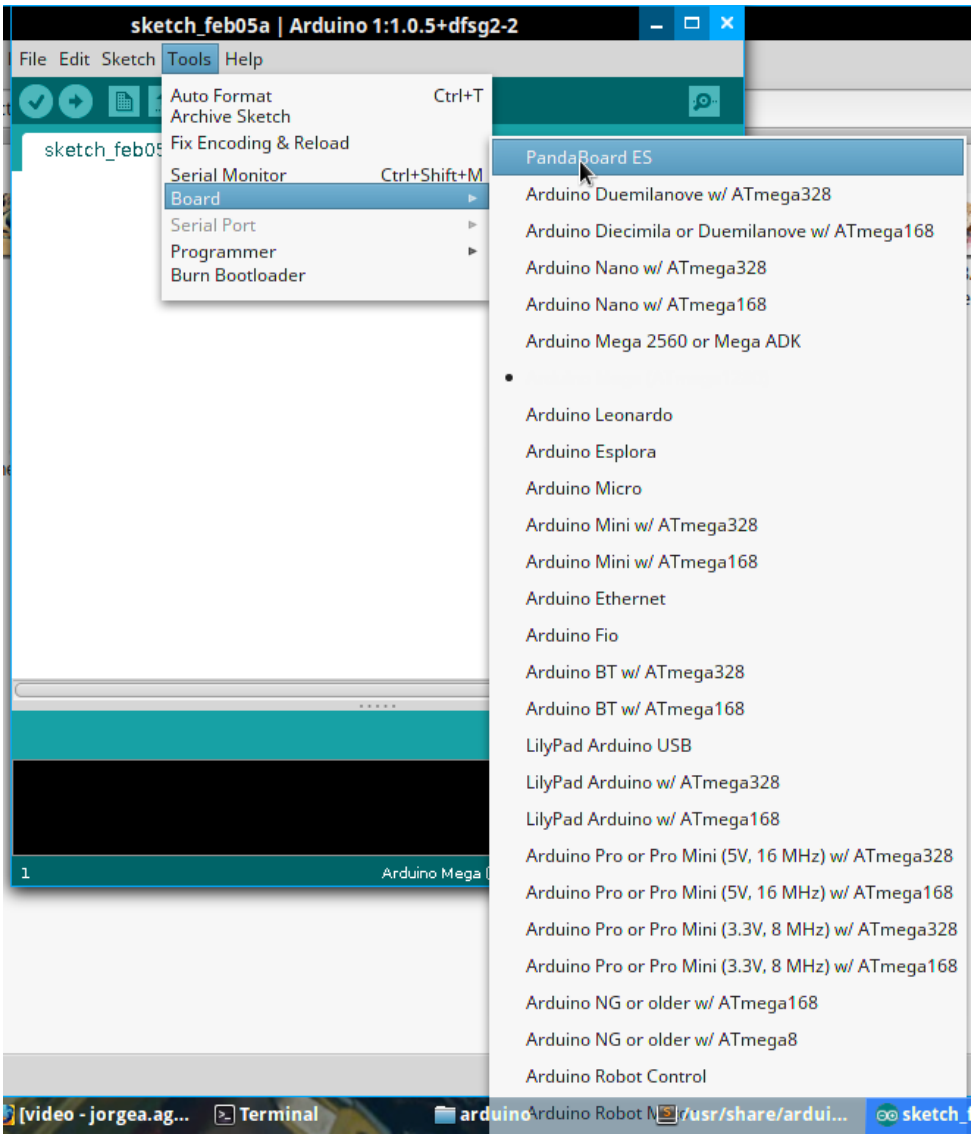

Figura 27. Pandaboard ES en el IDE de Arduino

Dentro del archivo mencionado se cambian los parámetros:

- name
- upload.maximum_size
- build.mcu
- build.f_cpu
- build.core
- build.variant

4.3.2 Cores GPIOS

La figura 28 ilustra el contenido del directorio core, en el cual se encuentran todas las dependencias disponibles en el IDE de Arduino para el desarrollo de aplicaciones y programas de un determinado microcontrolador como por ejemplo de la serie ARM, SAM o para este caso dirigido a la Pandaboard:

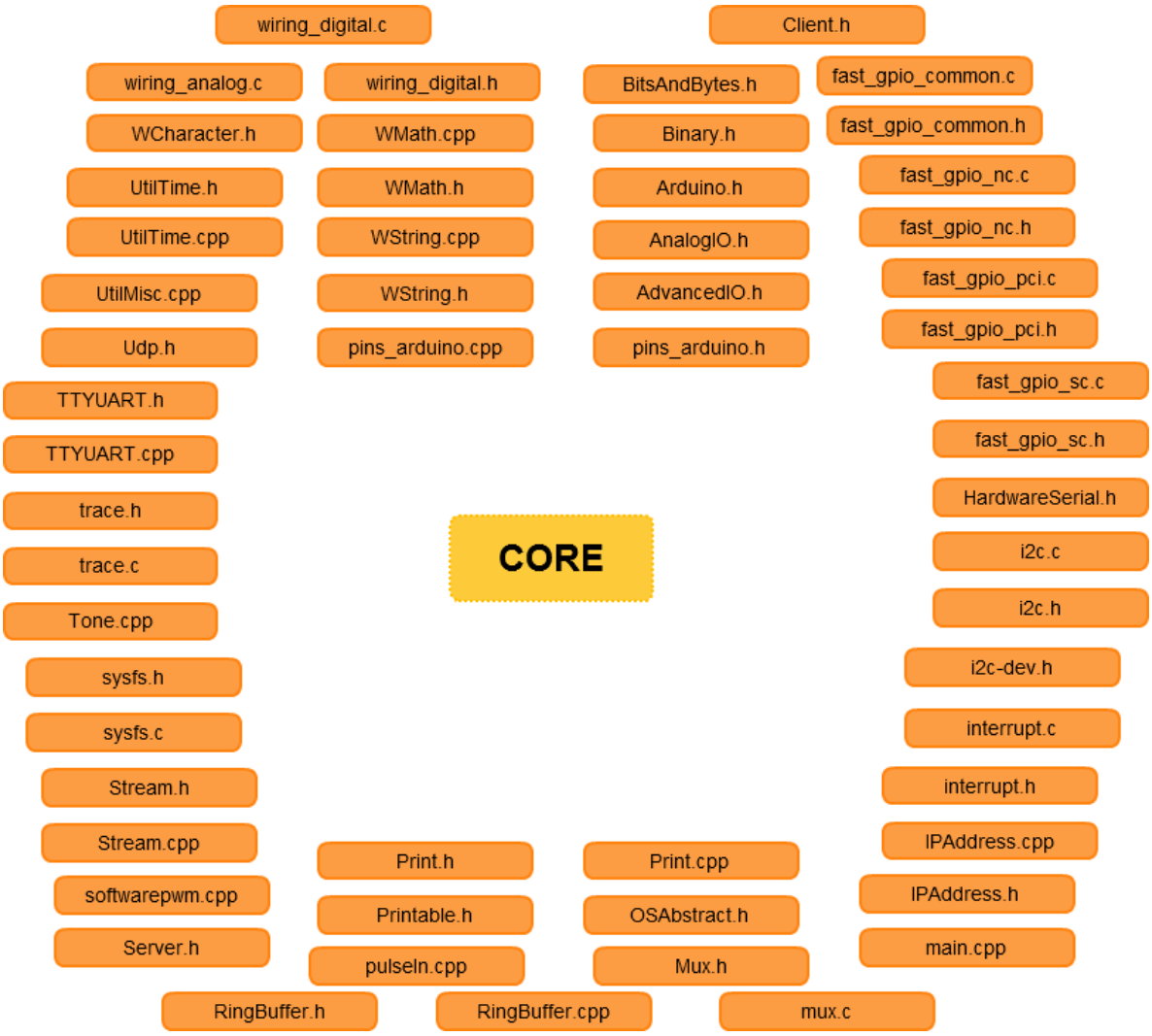

Figura 28. Componentes del Core de Arduino

La tabla 11 describe la funcionalidad del archivo Core Arduino:

Componente	Funcionalidad
AdvancedIO.h	Funciones para manejo de IO avanzadas.
AnalogIO.h	Definición de cabeceras para implementación analógica.
Arduino.h	Cabecera principal, con prototipos de funciones para IDE.
binary.h	Definiciones para las constantes binarias.
BitsAndBytes.h	Funciones de manipulación de bits.
Client.h	Interfaz de conexión abstracta.
fast_gpio_common.c	Utilidades para funciones rápidas de entrada y salida.
fast_gpio_common.h	Definición de utilidades para funciones rápidas de entrada y salida.
fast_gpio_nc.c	Implementar un camino GPIO rápido para el North-Cluster.
fast_gpio_nc.h	Definición para implementar un camino GPIO rápido para el North-Cluster.
fast_gpio_pci.c	Implementa ruta GPIO rápida.
fast_gpio_pci.h	Prototipos de funciones de GPIO rápidas.
fast_gpio_sc.c	Implementar un camino GPIO rápido para el South-Cluster.
fast_gpio_sc.h	Definición para implementar un camino GPIO rápido para el South-Cluster.
HardwareSerial.h	Librería de hardware serial de cableado.
i2c.c	Librería para i2c.
i2c.h	Definición de librería para i2c.
i2c-dev.h	Controlador i2c-bus, interfaz de dispositivo char.
interrupt.c	Interfaz de interrupción.
interrupt.h	Definición de interfaz de interrupción.
IPAddress.cpp	Métodos para manejar y pasar alrededor de las

	direcciones IP.
IPAddress.h	Cabecera de definiciones del archivo IPAddress.cpp.
main.cpp	Bucle principal espacio de usuario para las placas de la familia Intel Galileo.
mux.c	Alto nivel de abstracción para muxing a través de las tarjetas.
Mux.h	Definición de abstracción de alto nivel para muxing a través de las tarjetas.
OSAbstract.h	Conjunto de funciones principales para userpace.
Print.cpp	Clase base que provee print() y println().
Print.h	Clase base que provee las definiciones para print() y println().
Printable.h	Clase interfaz que permite la impresión de tipos complejos.
pulseIn.cpp	Librería pulseIn para Intel Galileo.
RingBuffer.cpp	Librería de ring buffer para propósito general.
RingBuffer.h	Definición de constantes y variables para buffering de los datos seriales de entrada.
Server.h	Cabecera de la librería Ethernet.
softwarepwm.cpp	Librería para producir señales PWM sobre algún pin arbitrario.
Stream.cpp	Añade métodos para transmitir de la clase stream.
Stream.h	Clase base para los flujos basados en caracteres.
sysfs.c	Librería para manipular el estado de los pines GPIO.
sysfs.h	Definición de las condiciones de estado en sysfs.c.
Tone.cpp	Librería para puerto Tone.
trace.c	Librerías para debuging.
trace.h	Define funciones de traces de bajo nivel
TTYUART.cpp	Implementación para comunicación TTY.
TTYUART.h	Definición de implementación para comunicación TTY

Udp.h	Librería para enviar y recibir paquetes UDP.
UtilMisc.cpp	Utilidades y funciones para intel Galileo.
UtilTime.cpp	Proporciona funciones para tiempo.
UtilTime.h	Proporciona definiciones para funciones para tiempo.
WCharacter.h	Conjunto de caracteres para funciones de cableado y Arduino.
wiring_analog.c	Funciones de entrada y salida analógica.
wiring_digital.c	Funciones de entrada y salida digital.
wiring_digital.h	Describe una breve configuración de pines para que se comporten como entrada o como salida.
WMath.cpp	Librería matemática personalizada para Arduino.
WMath.h	Definición de archivo WMath.cpp.
WString.cpp	Librería string para funciones de cableado y Arduino.
WString.h	Definición de librería string para funciones de cableado y Arduino.
pins_arduino.h	Archivo que contiene la definición de funciones de los pines de Arduino.
pins_arduino.cpp	Contiene la descripción e implementación de los pines dividiéndolos en digitales, analógicos, así como también los dedicados a la transmisión y recepción vía serial.

Tabla 11. Funcionalidades del Archivo Core Arduino.

4.3.3 Librerías WiFi

Esta librería permite a la board Arduino conectarse a internet. Puede servir como un servidor que acepta conexiones entrantes o un cliente que realiza los salientes. La librería soporta WEP y WPA2 Personal encryption, pero no WPA2 Enterprise. También se debe tener en cuenta que si el SSID no se transmite, el shield no se puede conectar. Arduino se comunica con el shield WiFi usando el bus SPI [84]

La librería WiFi es muy similar a la librería Ethernet, y muchas de sus funciones se llaman igual.

Para este aparte es necesario también usar la librería SPI (Serial Peripheral Interface) es un protocolo de datos serial síncrono utilizado por los microcontroladores para comunicarse con uno o más dispositivos periféricos rápidamente en distancias cortas. Con una conexión SPI siempre hay un dispositivo master (usualmente un microcotrlador o tarjeta de desarrollo) que controla a los dispositivos periféricos slave. Tipicamente hay tres líneas comunes a todos los dispositivos [85]:

- MISO (Master In Slave out) - La línea de esclavo para el envío de datos al maestro
- MOSI (Master Out Slave In) - La línea principal para el envío de datos a los periféricos
- SCK (Serial Clock) - Los impulsos de reloj que sincronizan la transmisión de datos generada por el maestro.

Dentro de la librería de WiFi existe un directorio que contiene unos ejemplos, algunos de los cuales son útiles en este estudio:

- ScanNetwork: Este ejemplo imprime la dirección MAC del shield de WiFi para la Arduino y escanea las redes WiFi disponibles usando dicho shield. Este escaneo se realiza cada 10 segundos. Esta aplicación actualmente no conecta a ninguna red dado que no tiene ningún esquema de encriptado especificado. Se espera que funcione de igual manera en la Pandaboard ya que esta cuenta con un elemento hardware integrado para WiFi [86].

- WiFiSendReceiveUDPString : Este sketch (nombre que usa Arduino para los programas) espera por un paquete UDP sobre un puerto local. Cuando se

valida que el paquete ha sido recibido, un paquete de reconocimiento es enviado de vuelta al cliente sobre un especifico puerto de salida [87].

4.4 Comunicación PC-Tarjeta.

SSH-NFS

SSH (o *S*ecure *SH*ell) es un protocolo para comunicación segura entre dos sistemas usando una arquitectura cliente/servidor que permite conectarse a un dispositivo remotamente a través del uso de un intérprete de comandos.

Un programa relacionado, el scp, reemplaza otros programas diseñados para copiar archivos entre hosts como rcp que son menos seguros dada su antigüedad y falta de encriptación de contraseñas entre el cliente y el servidor [88].

Para esta parte inicialmente se usaba ssh en la comunicación, pero en el cambio de sistema operativo también se utilizó nfs. Es por eso que a continuación se describirán los procesos alrededor de estos y las dificultades encontradas.

En el sistema Debian se instala el servicio ssh como cliente (openssh-client) y en el pc el servidor (openssh-server,openssh-sftp-server) de la siguiente manera:

apt-get install ssh
apt-get install openssh-client
apt-get install openssh-server

Para comunicarse desde el pc se ejecuta el siguiente comando:

ssh -p 22 debian@direccion-ip-de-la-tarjeta

Acto siguiente se pedirá confirmación para vincular dicha dirección IP y la contraseña correspondiente al usuario debian en la tarjeta.

Para copiar los ejecutables hasta la panda se hace con un comando ligeramente diferente (ver figura 29):

scp -P 22 /ruta-de-los-ejecutables debian@direccion-ip-de-la-tarjeta:ruta-a-donde-copiar

Igualmente se debe agregar la contraseña para poder realizar la acción. En el caso de copiar directorios completos se debe agregar '-r'.

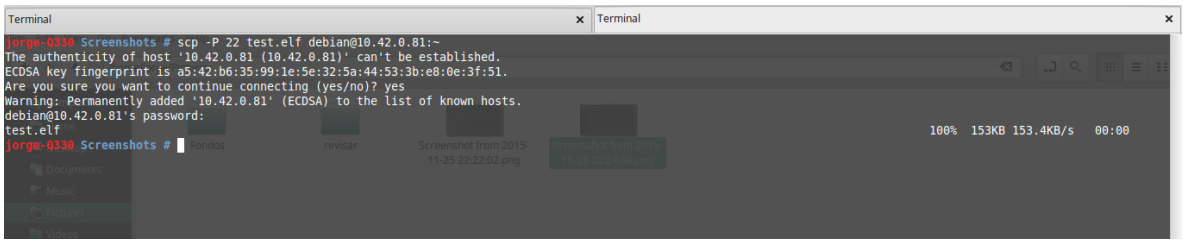

Figura 29. Enlace de ejecutable con la tarjeta con SSH

Para el sistema Linaro, a pesar de que tenía instalado ssh por defecto no se pudo utilizar debido a que ninguno de los passwords encontrados servía; por lo que se recurre al método para compartir carpetas vía nfs. El Network File System *(Sistema de archivos de red)*, o NFS es un protocolo de red que permite acceder a archivos en dispositivos remotos de la misma forma como si estuvieran localmente con un modelo cliente/servidor [88], ver figuras 30 y 31.

Se procede a instalar el nfs [90] en el servidor (host) y en el cliente (tarjeta) montando las correspondientes carpetas:

apt-get install nfs-kernel-server portmap (servidor)
apt-get install nfs-common (cliente)

ipServidor -> 192.168.190.70

ipCliente -> 192.168.190.11

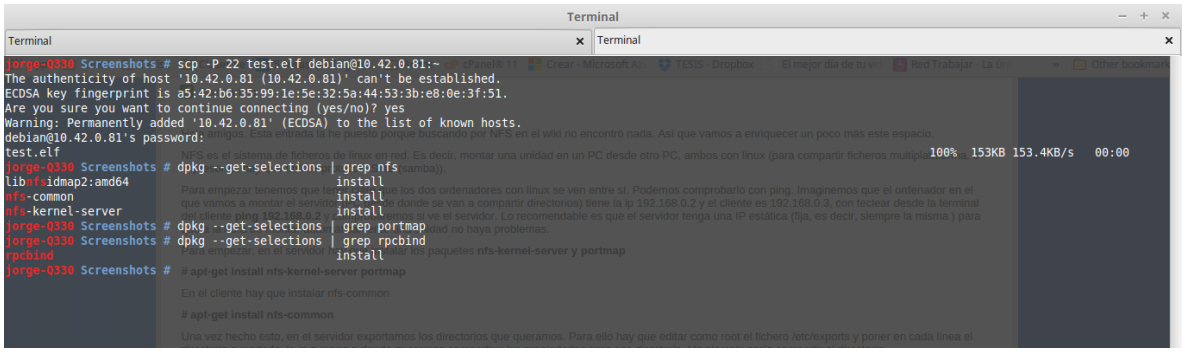

Figura 30. Confirmación paquetes NFS y rpcbind (portmap) en el host

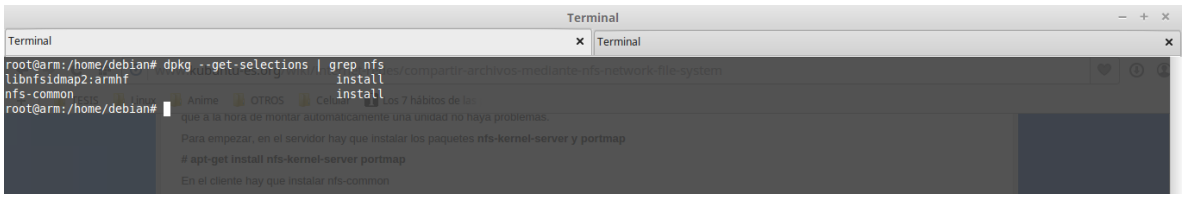

Figura 31. Confirmación de paquete NFS en la tarjeta

Hay que editar como root el fichero /etc/exports y poner en cada línea el directorio exportado, la IP o rango a donde queremos compartir y las propiedades para ese directorio

/home/jorge/Desktop/Ejecutables/ 192.168.190.11/255.255.255.0

Así se exporta el directorio con lectura, escritura, sincronización y visualización de subdirectorios. En este apartado también son importantes los permisos de los propios directorios. Cabe destacar que 255.255.255.0 es la máscara de red que en caso de tener subredes es importantísimo, ver figura 32.

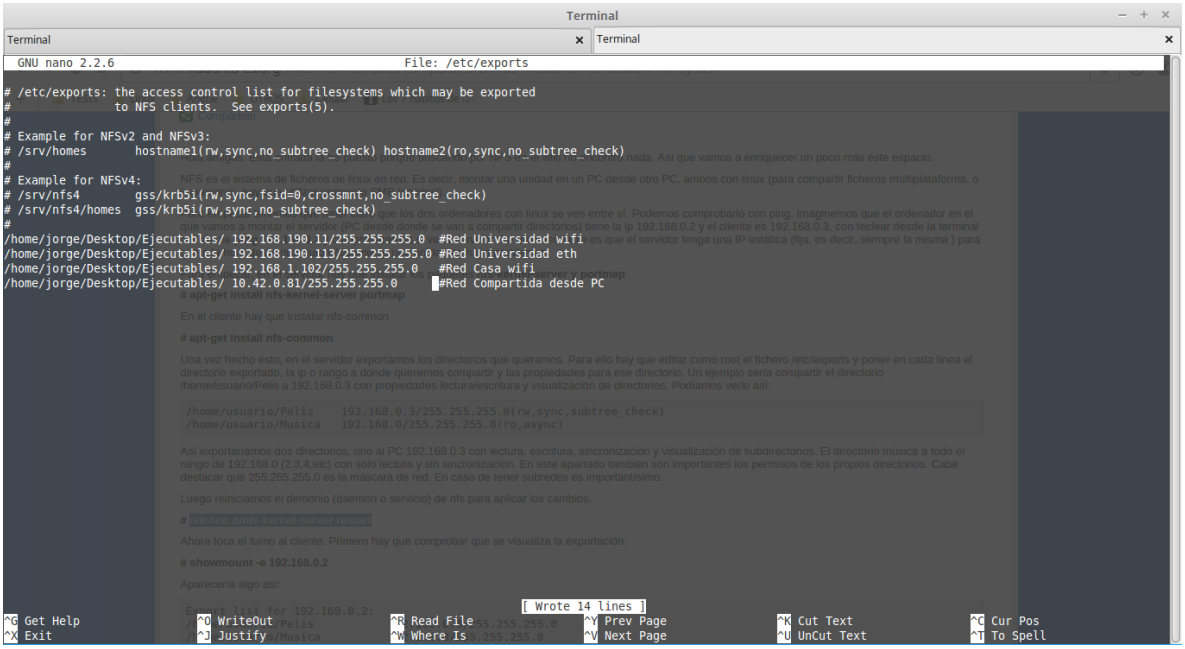

Figura 32. Configuración archivo /etc/exports

Se reinicia el demonio (daemon o servicio) de nfs (figura 32.) para aplicar los cambios:

 # /etc/init.d/nfs-kernel-server restart

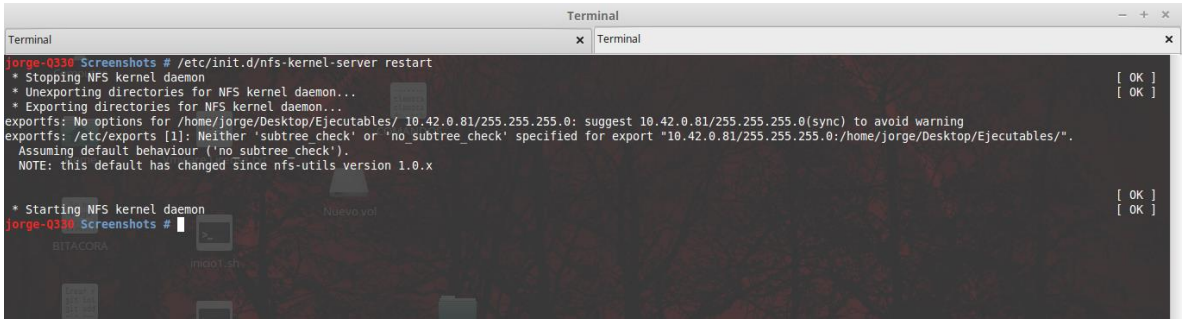

Figura 33. Reinicio del servicio NFS

Ahora es el turno para el cliente. En primer lugar hay que comprobar que se visualiza la exportación con el siguiente comando:

 # showmount -e 192.168.190.70

Aparece:

Export list for 192.168.190.70:
/home/jorge/Desktop/Ejecutables 192.168.190.11/255.255.255.0

Este resultado se puede ver en la figura 34:

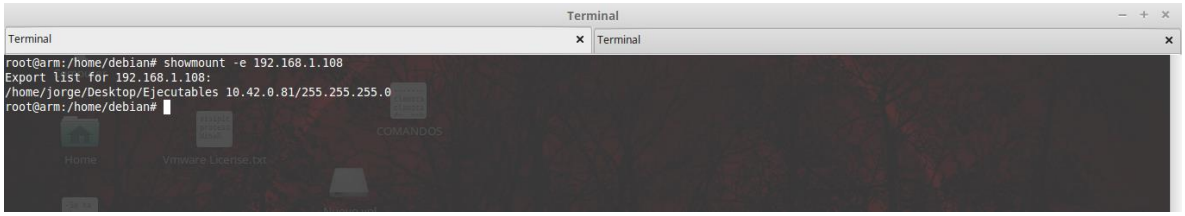

Figura 34. Prueba de visualización de fichero exportado en la tarjeta

Luego se debe crear los directorios donde se van a montar estas unidades:

$ mkdir /home/linaro/Ejecutables-Servidor

Y se monta con:

 # mount -t nfs 192.168.190.70:/home/jorge/Desktop/Ejecutables /home/linaro/Ejecutables-Servidor

Aparece el error "mount.nfs: access denied by server while mounting 192.168.190.70:/home/jorge/Ejecutables"

Se instala el paquete iptables en la tarjeta ya que no lo tenía (cortafuegos utilizado para gestionar las conexiones en Linux) y se intenta instalar portmap en el servidor; en el proceso aparece el mensaje "Note, selecting 'rpcbind' instead of 'portmap'" indicando que ya tiene instalada una herramienta para este propósito (conocer los servicios que corren en una máquina objetivo).

Al intentar montar la carpeta en el cliente nuevamente aparece un problema al no poder ubicarla; pero al volver a correr el comando la carpeta finalmente se monta y se pasan los ejecutables para correrlos en la tarjeta, incluyendo el archivo test.elf de la compilación con las librerías.

En la SD_3 se instala nfs y se sigue el procedimiento anterior con punto de montaje en:

mount -t nfs 192.168.1.109:/home/jorge/Desktop/Ejecutables /home/linaro/Ejecutables-Servidor

Y funciona sin ningún problema.

.

Capítulo 5

5 Validación Mediante un Caso de Estudio del mecanismo de Integración de la Pandaboard en el Ambiente de Arduino

5.1 Plan de pruebas.

El plan de pruebas que se presenta a continuación se divide en dos partes, la primera es la de planeación del experimento donde se describe el objetivo de las mismas. La segunda, corresponde al plan de ejecución, donde se explica cómo es el proceso de recolección de datos.

5.1.1 Descripción del experimento

El experimento está basado en el caso de estudio, el cual consiste en que la Pandaboard va a recolectar una serie de datos suministrados por una red de sensores. En este punto la tarjeta se va a comportar como un servidor TCP o UDP de tal manera que recibirá la información vía red WiFi. Acto seguido la Pandaboard va a enviar la información recolectada también como servidor TCP o UDP hacia otro servidor pero vía Ethernet, según se muestra en la figura 35:

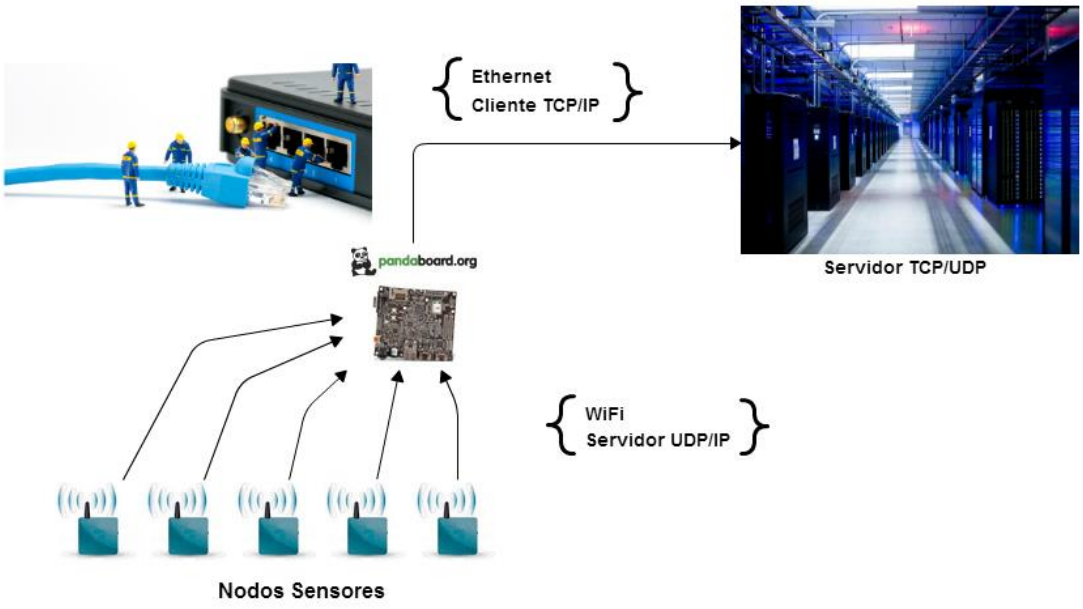

Figura 35. Diagrama del Caso de Estudio

Definición Objetivo: medir el tiempo que tarda un usuario en crear un servidor TCP o UDP tanto para la red WiFi como para Ethernet y desarrollar un programa que recopile información de una serie de sensores hacia la Pandaboard por medio de WiFi y luego los envíe desde la tarjeta hacia un servidor vía Ethernet; todo esto usando el servidor creado previamente. Para ello los desarrolladores del experimento suministrarán una guía con el objetivo del experimento; para ello deberán utilizar tanto la adaptación en Arduino como el ambiente embebido de Linux en el computador a través del uso de *cu* como medio de comunicación con la tarjeta, para finalmente evaluar que tan intuitivo perciben los usuarios el uso del IDE de Arduino con porting para la Pandaboard con relación al otro método.

Formulación de Hipótesis:

H0 (Hipótesis Nula): el tiempo de creación de un servidor TCP o UDP y el desarrollo de un programa para recepción y envío de datos con la Pandaboard utilizando el IDE de Arduino con porting o con otro método NO varía de forma considerable.

Ha (Hipótesis alternativa): el tiempo de creación de un servidor TCP o UDP y el desarrollo de un programa para recepción y envío de datos con la Pandaboard utilizando el IDE de Arduino con porting varía de forma considerable en comparación al otro método.

Selección de variables:

- Independiente.
- Dependiente.

Selección de sujetos: El experimento se desarrolla con 6 estudiantes de la Universidad del Cauca (estudiantes de la facultad de Ingeniería Electrónica y Telecomunicaciones de los últimos semestres y algunos egresados) los cuales tienen conocimientos sobre el manejo de Arduino, comandos en Linux y creación de servidores TCP/UDP.

Objetos experimentales: se desea conocer el tiempo de creación de un servidor TCP/UDP y el desarrollo de programas en el lenguaje C y C++ utilizando dos vías: la primera desde el ambiente embebido de Linux para Pandaboard a través de comunicación por puerto serial con la línea de comandos “cu”; y la segunda usando un IDE Arduino al que previamente se le ha realizado el porting para la Pandaboard. También se quiere conocer las diferentes opiniones de los sujetos del experimento respecto a los dos métodos de desarrollo.

Se tuvieron en cuenta unas condiciones generales para todos los experimentos, las cuales se describen a continuación:

- Se escogieron tres (3) sujetos (de 6) al azar para cada prueba, los cuales han sido invitados a ingresar a una sala, en la cual se les puso el objetivo de crear un servidor y de desarrollar un programa basándose en una guía (Anexo C) que se entrega al entrar a la prueba. Antes del inicio del test se les informó en

qué consiste la prueba que debían desarrollar, sin embargo, es hasta que ingresan al laboratorio, que fueron notificados de las herramientas software que iban a utilizar para desarrollar dicho programa.

- La guía de desarrollo entregada al inicio, describe cómo utilizar cada una de estas herramientas, y la manera para compilar y ejecutar los programas sobre la Pandaboard.

- El tiempo comienza a correr cuando el sujeto inicia la creación del programa. Esto no se le notifica al sujeto (para evitar presiones y ansiedad como factores que pudieran afectar el resultado).

- Si se presenta algún error durante la prueba el cronómetro es detenido hasta que el problema sea resuelto. Una vez logrado esto se activa nuevamente el cronómetro.

- El tiempo se detiene cuando el sujeto compila y envía el programa a la tarjeta. Finalmente, con el reloj detenido, se realiza una prueba de funcionamiento con el ejecutable generado.

Una vez finalizada la primera parte los sujetos intercambiarán las pruebas pero esta vez no se les medirá el tiempo sino que al finalizar se les realizará una encuesta para medir su percepción respecto a la creación del servidor y al desarrollo de un programa para la Pandaboard utilizando el IDE de Arduino.

5.1.2 Ejecución de Pruebas.

El experimento se dividió principalmente en dos pruebas: **P1** y **P2**; a continuación se explican cada una de ellas

- **P1**: se formaron dos grupos de siete personas al azar; cada grupo se etiquetó como: “Prueba A” y “Prueba B”, los cuales siguieron las guías mencionadas

previamente. El primer grupo utilizó la herramienta software IDE de Arduino y el segundo grupo utilizó el editor de texto Sublime para desarrollar el código, el compilador cruzado para *armhf* ejecutado desde la terminal de Linux con el uso de un archivo *makefile*, el programa *ssh* para enviar el ejecutable remotamente a la tarjeta y los comandos de *cu* para conectarse con esta a través de puerto serial para la parte de despliegue.

- **P2**: la metodología es similar a la utilizada para P1, la diferencia ésta en que se invirtieron los sujetos de la muestra a fin de que realizaran la prueba contraria a la ejecutada en la parte anterior. En esta prueba la evaluación se realizó por medio de una encuesta a fin de medir el grado de afinidad y practicidad para cada método según los sujetos evaluados.

El público que participó en el experimento fueron seis (6) estudiantes de la facultad de Ingeniería Electrónica y Telecomunicaciones y egresados que ya habían realizado la primera parte del experimento.

El objetivo de la encuesta es conocer la opinión de los encuestados sobre el manejo de la Pandaboard desde el IDE de Arduino en comparación al uso convencional desde la terminal para Pandaboard.

5.2 Resultados obtenidos.

En esta sección se muestran las gráficas con los resultados obtenidos para las dos pruebas.

Las figuras 36, 37, 38 y 39 muestran los resultados de las pruebas P1 y P2 donde se midió el tiempo que los sujetos (s1, s2, s3, s4, s5 y s6) tardaron en desarrollar el programa y realizar las configuraciones en la Pandaboard para el caso de estudio con dos técnicas diferentes.

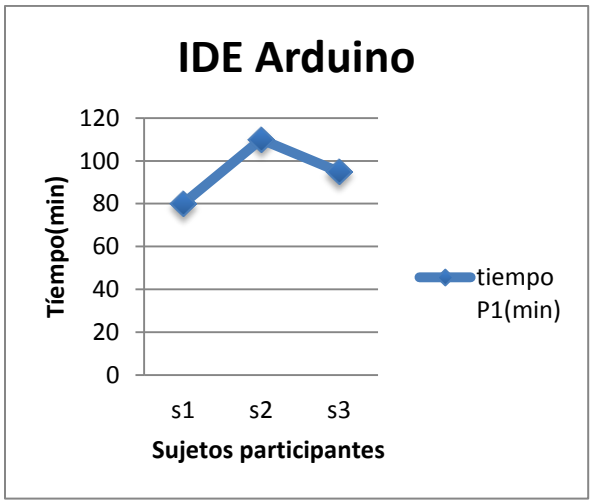

Figura 36. Resultado prueba P1 IDE Arduino para sujetos s1, s2 y s3.

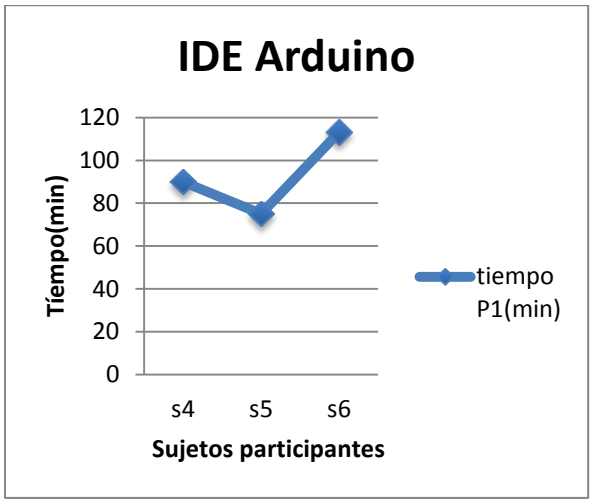

Figura 37. Resultado prueba P1 IDE Arduino para sujetos s4, s5 y s6.

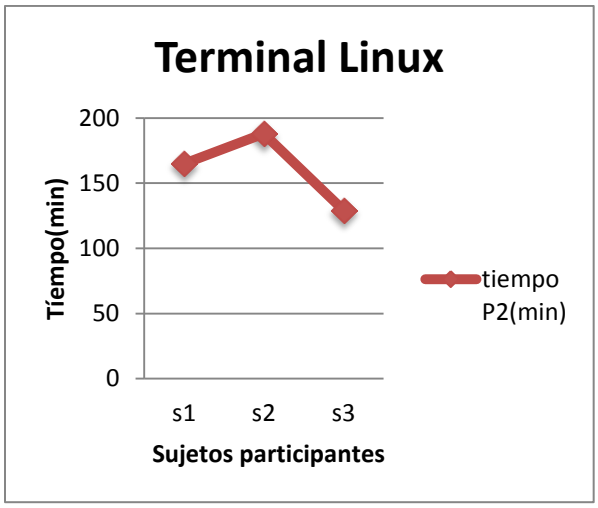

Figura 38. Resultado prueba P2 Terminal Linux para sujetos s1, s2 y s3.

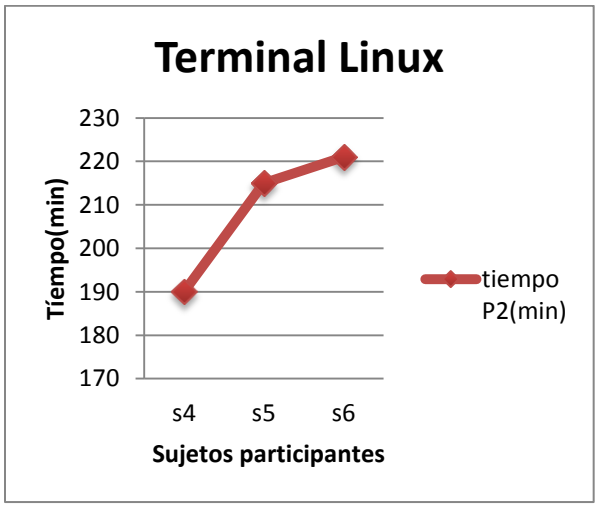

Figura 39. Resultado prueba P2 Terminal Linux para sujetos s4, s5 y s6.

Las figura 40 muestra un comparativo en los tiempos de desarrollo de los sujetos s1, s2 y s3 utilizando las dos técnicas en el mismo caso de estudio

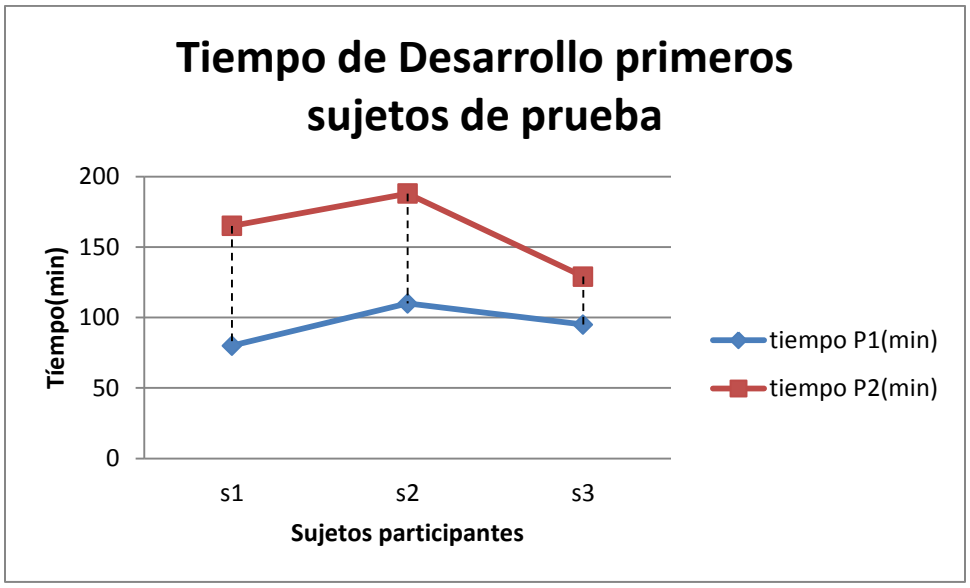

Figura 40. Comparativo de tiempos de los sujetos s1, s2 y s3.

Las figura 41 muestra un comparativo en los tiempos de desarrollo de los sujetos s4, s5 y s6 utilizando las dos técnicas en el mismo caso de estudio

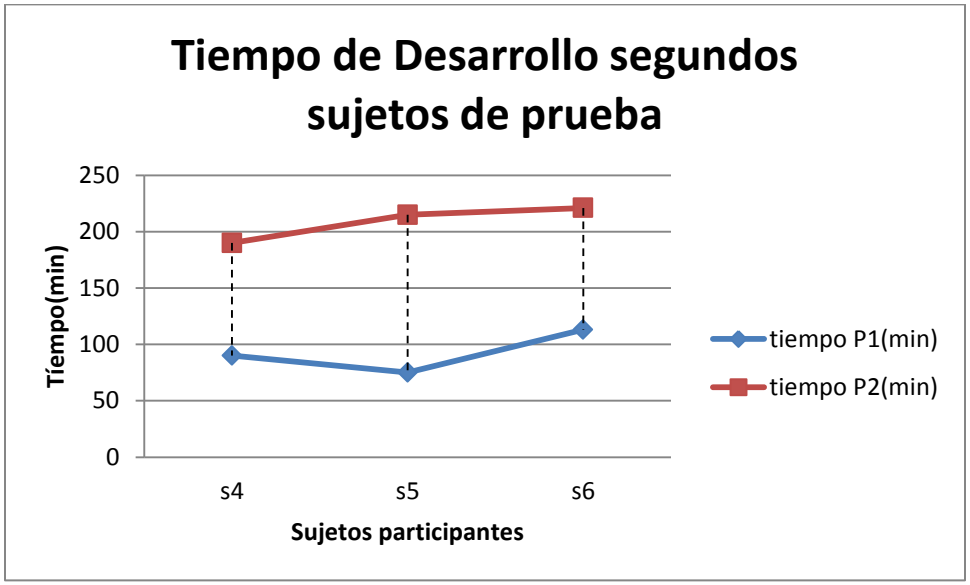

Figura 41. Comparativo de tiempos de los sujetos s4, s5 y s6.

Las gráficas anteriores muestran claramente que los tiempos de desarrollo con el IDE de Arduino fueron mejores que con el otro método, pues la diferencia es de hasta125 minutos en el mayor de los casos y de 34 minutos en el menor.

Las figuras 42, 43, 44, 45, 46 y 47 muestran los resultados de cada pregunta de la encuesta realizada al final de la prueba P2.

Pregunta 1: ¿Le pareció intuitivo programar con el primer método en comparación con el segundo?

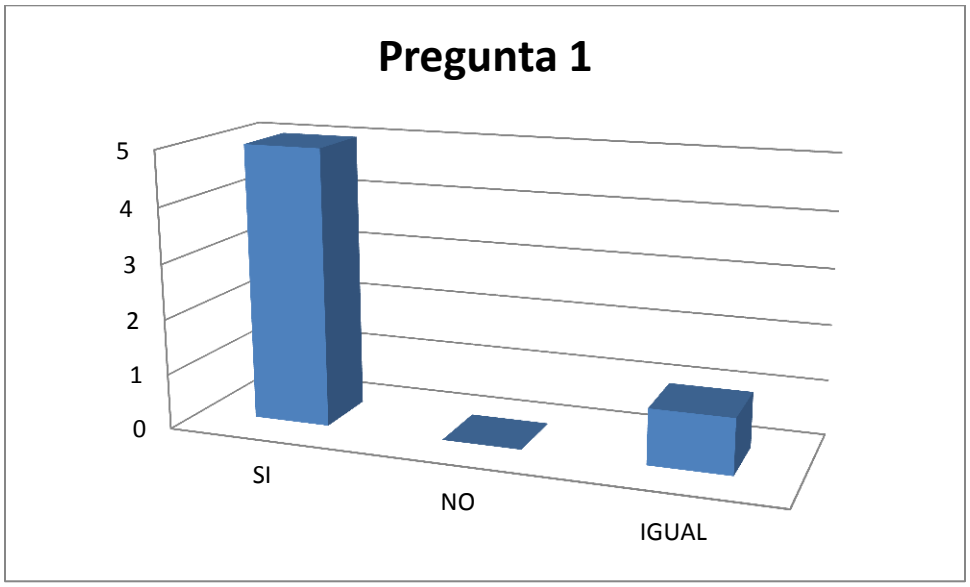

Figura 42. Resultados pregunta 1.

Pregunta 2: ¿Cuál método considera que es más rápido en cuanto a tiempos de desarrollo?

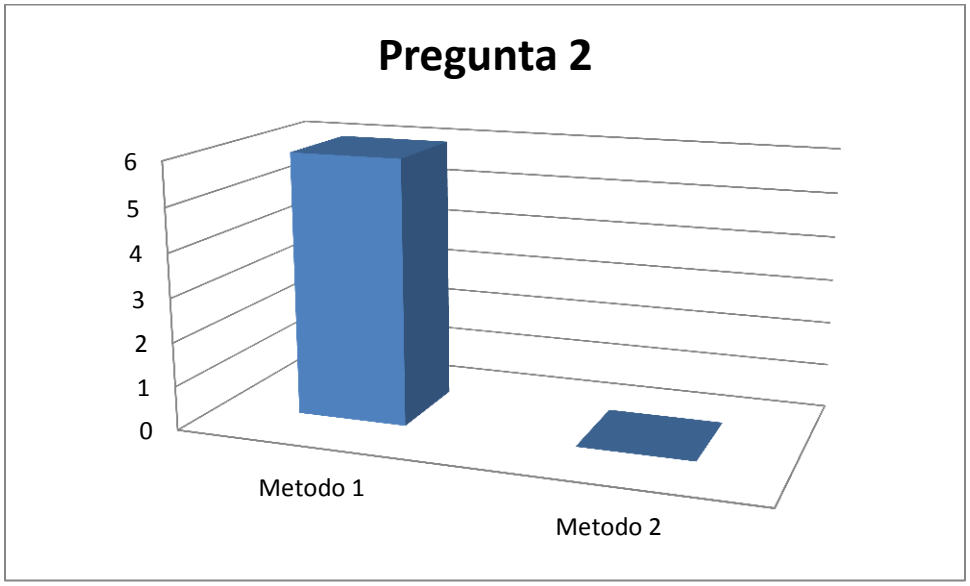

Figura 43. Resultados pregunta 2.

Pregunta 3: ¿El uso de una interfaz gráfica le hace más cómoda la tarea de programar?

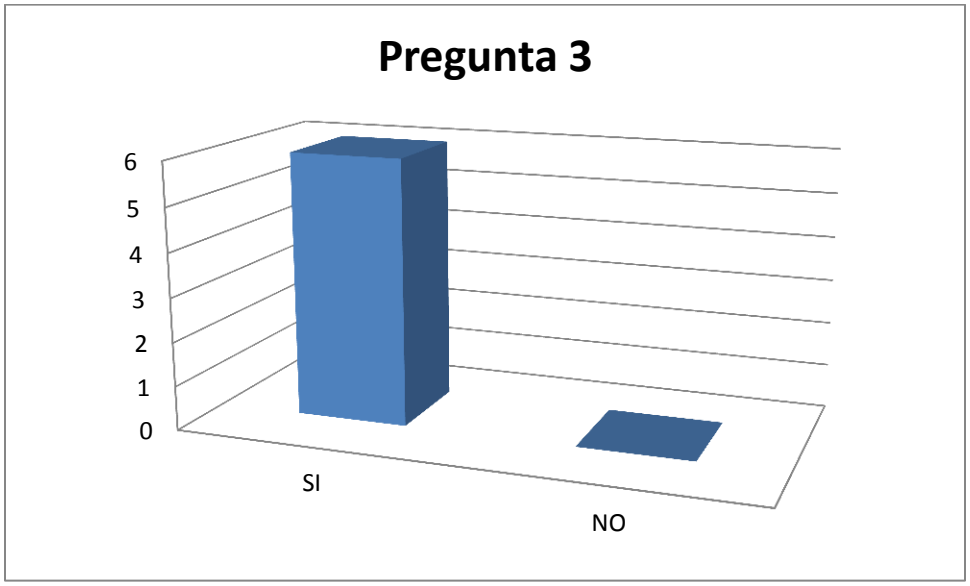

Figura 44. Resultados pregunta 3.

Pregunta 4: ¿Considera que el segundo método hace uso de demasiadas herramientas software para el proceso de desarrollo?

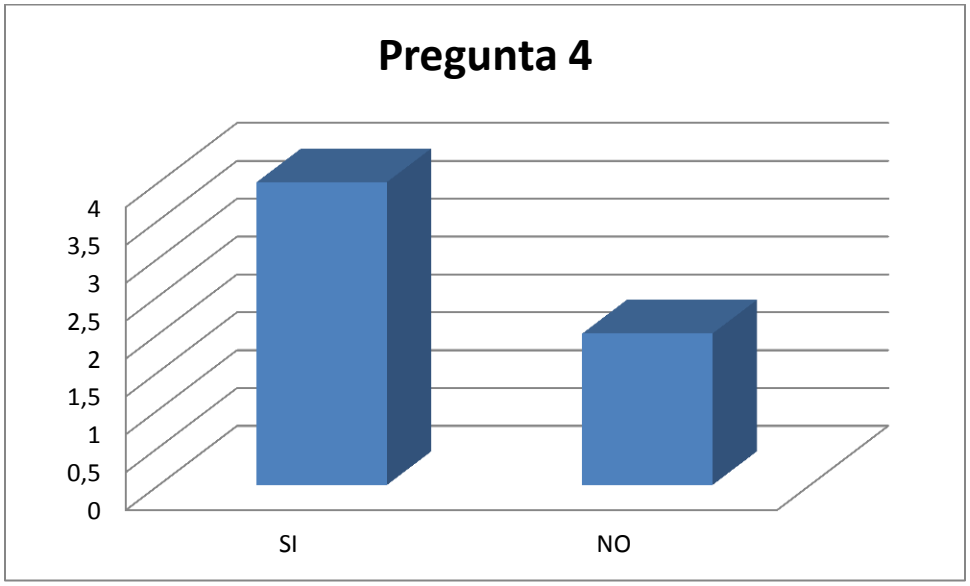

Figura 45. Resultados pregunta 4.

Pregunta 5: ¿Considera que el primer método podría mejorar o complementar la experiencia de desarrollo en la Pandaboard?

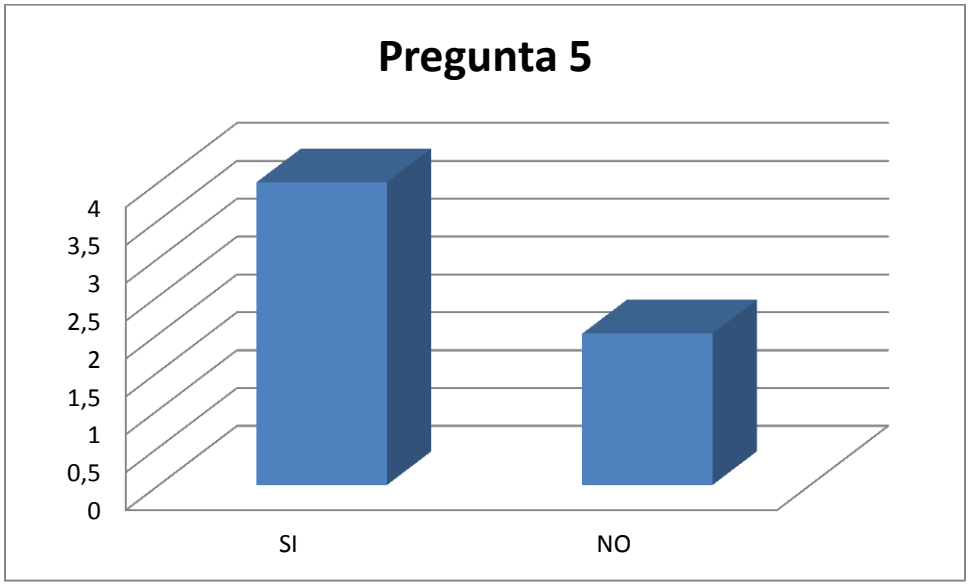

Figura 46. Resultados pregunta 5.

Pregunta 6: ¿Considera que el primer método motivaría a las personas a utilizar la Pandaboard en sus proyectos?

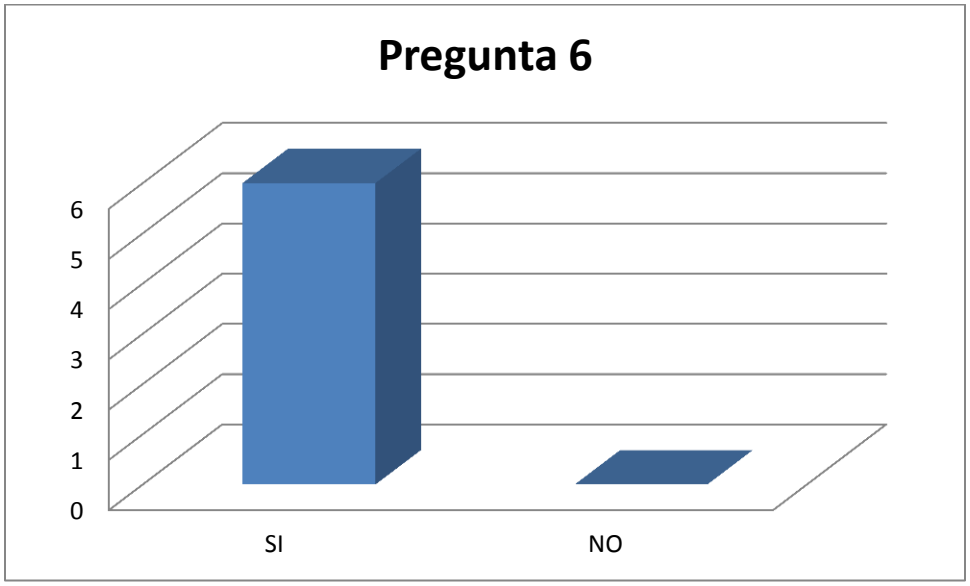

Figura 47. Resultados pregunta 6.

La pregunta número uno (1): ¿Le pareció intuitivo programar con el primer método en comparación con el segundo?, arroja que el total de los sujetos que participaron en la prueba están de acuerdo en que el uso de un IDE para programar en la Pandaboard es más cómodo a excepción de uno que está bastante familiarizado con las dos opciones.

La pregunta número dos (2): ¿Cuál método considera que es más rápido en cuanto a tiempos de desarrollo?, arroja que todos los participantes de la prueba consideran que es más ágil el desarrollo para la tarjeta usando esta adaptación del IDE

La pregunta número tres (3): ¿El uso de una interfaz gráfica le hace más cómoda la tarea de programar?, arroja que el total de los encuestados le parece más agradable la programación usando la interfaz gráfica del IDE de Arduino.

La pregunta número cuatro (4): ¿Considera que el segundo método hace uso de demasiadas herramientas software para el proceso de desarrollo?, arroja que el 66.6% de los sujetos participantes en la prueba considera que el número de herramientas utilizadas para el desarrollo del programa es numeroso.

La pregunta número cinco (5): ¿Considera que el primer método podría mejorar o complementar la experiencia de desarrollo en la Pandaboard?, arroja que sólo el 33.3% de los participantes en las pruebas consideran que el uso de este método no contribuiría a añadir un beneficio en cuanto a la experiencia de desarrollo sobre esta board.

La pregunta número seis (6): ¿Considera que el primer método motivaría a las personas a utilizar la Pandaboard en sus proyectos?, arroja que la totalidad de los sujetos de prueba piensan que esta integración de la Pandaboard con el IDE de Arduino podría generar un mayor uso de esta tarjeta en proyectos de desarrollo.

5.3 Conclusiones de la validación

- La opinión de los sujetos encuestados muestra que es más cómodo e intuitivo el uso de un IDE para el desarrollo sobre la Pandaboard.

- Para que el tiempo sea relativamente despreciable entre el uso de los dos distintos métodos, el usuario o programador debe tener experiencia o un conocimiento más amplio de Linux y del manejo del IDE de Arduino.

- El uso del IDE para el desarrollo sobre la Pandaboard mejora los tiempos de programación de las aplicaciones para el usuario promedio.

Capítulo 6

6 Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones, resultado del desarrollo de este trabajo de investigación como también las propuestas para trabajos futuros que se podrían realizar teniendo como base el presente trabajo.

6.1 Conclusiones.

- Se encontró una carencia de documentación y trabajos que estuvieran relacionados con la Pandaboard.

- Arduino mejora el desempeño de los desarrolladores para la Pandaboard.

- La interfaz de Arduino presenta una alternativa más intuitiva al momento de desarrollar una aplicación en comparación al método con comandos de Linux, dado que para este se hacen necesarios no solamente conocimientos de programación, sino también, una experiencia al menos mínima en el manejo de este tipo de comandos y de este sistema operativo.

- La integración de las diversas herramientas involucradas en el proceso de compilación y enlace sin hacer uso del IDE de Arduino es nula haciendo de esta no una tarea difícil pero si tediosa.

6.2 Trabajo Futuro.

- Automatizar la ejecución del binario enviado a la tarjeta, para que este se ejecute desde el arranque.

- Hacer que el enlace y la compilación de los códigos para la Pandaboard se realicen desde el IDE de Arduino.

- Hacer más pruebas, con un número de personas mayor y con códigos de mayor complejidad.

- Trabajar con otras librerías y recursos de Arduino para ampliar los procesos que se puedan realizar sobre la Pandaboard.

- Probar la integración de la Pandaboard con otros entornos de desarrollo.

7 Bibliografía.

[1] Internet of Things, [Online]. Disponible en: http://www.cisco.com/web/solutions/trends/iot/overview.html [Consultado: Junio 13, 2014]

[2] Internet de los Objetos, [Online]. Disponible en: http://www.itu.int/itunews/manager/display.asp?lang=es&year=2005&issue=09&ipage=things&ext=html [Consultado: Junio 12, 2014]

[3] D. Del Peral, “Integración continua para open hardware”, tesis pregrado, Universidad Carlos III de Madrid, Madrid, España, 2012.

[4] Arduino, [Online]. Disponible en: http://www.arduino.cc/ [Consultado: Junio 17, 2014]

[5] PandaBoard: is Open OMAP™ 4 mobile software development platform, [Online]. Disponible en: http://pandaboard.org/node/300/#specs [Consultado: Junio 17, 2014]

[6] S. Shue, “Low cost semi-autonomous sentry robot”, in Proceedings of IEEE Southeastcon, Orlando, FL, pp. 1-5, March 2012.

[7] I. Calabrese, “Embedded systems for prototyping underwater acoustic networks: The DESERT Underwater libraries on board the Pandaboard and NetDCU”, in Oceans, Hampton Road, VA, pp. 1-8, October 2012.

[8] M. Cerny, “Eye tracking system on embedded platform”, in International Conference on Applied Electronics (AE), Pilsen, pp. 51-54, September 2012.

[9] M. Shah, A. Sheth, and S. Shah, “Home Automation Using Embedded Web server”, 2012-13. [Online]. Available: http://www.spit.ac.in/wp-

content/uploads/profkttalele/project/2011-12/Home%20Automation/HomeAutomation.pdf

[10] C. Berger, A. Al Mamun, and J. Hansson, “COTS-Architecture with a Real-Time OS for a Self-Driving Miniature Vehicle”, presented at Workshop ASCoMS of the 32nd International Conference on Computer Safety, Reliability and Security, Toulouse, France, 2013.

[11] J. Wang, “Ballbot: A Low-Cost Robot for Tennis Ball Retrieval”, Electrical Engineering and Computer Sciences University of California at Berkeley, Berkeley, CA, USA, Tech. Rep. No. UCB/EECS-2012-157, June 2012.

[12] G. Pratyusha, and N.V. Ramesh, “PORTING THE LINUX KERNEL TO AN ARM BASED DEVELOPMENT BOARD”, International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 2, pp.1614-1618, Mar-Apr 2012.

[13] C. E. Serrano, Modelo Integral para el Profesional en Ingeniería, Popayán, Universidad del Cauca, 2 ed, 2008.

[14] C. E. Serrano, "Modelo para la Investigación Documental", Modelo Integral para el Profesional en Ingeniería, Popayán, Universidad del Cauca, 2 ed, pp. 12-20, 2008.

[15] C. E. Serrano, "Modelo para la Construcción de Soluciones", Modelo Integral para el Profesional en Ingeniería, Popayán, Universidad del Cauca, 2 ed, pp. 43 - 58, 2008.

[16] Internet de las cosas: concepto y ecosistema, [Online]. Disponible en: http://colombiadigital.net/actualidad/articulos-informativos/item/7821-internet-de-las-cosas-concepto-y-ecosistema.html [Consultado: Junio 1, 2015]

[17] ¿Qué es y cómo funciona el Internet de las cosas? [Online]. Disponible en: http://hipertextual.com/archivo/2014/10/internet-cosas/ [Consultado: Junio 1, 2015]

[18] CLÚSTER ICT-AUDIOVISUAL DE MADRID, Internet de las cosas:
Objetos interconectados y dispositivos inteligentes. [en línea]. Madrid: Marzo, 2013. Introducción. [citado el 1 de junio de 2015]. Disponible en internet: https://actualidad.madridnetwork.org/imgArticulos/Documentos/635294387380363206.pdf

[19] Executive Summary, “ITU Internet Reports 2005: The Internet of Things”, International Telecommunication Union (ITU), Geneva, Suiza, Tech. Rep, Noviembre 2005.

[20] D. Evans, “Internet de las cosas: Cómo la próxima evolución de Internet lo cambia todo”, Cisco Systems, Inc., San José, CA, Inf. Téc., Abril 2011.

[21] G. Gastaldi, J.A. Rapallini, H.O. Pascual, "Evaluación de programas de cálculo en ingeniería como herramienta de desarrollo para Codiseño Hardware / Software". IWS2002 VIII Workshop IBERCHIP, Guadalajara, México, Abril 2002.

[22] Hardware Libre, [Online]. Disponible en: http://www.ecured.cu/index.php/Hardware_libre [Consultado: Marzo 3, 2014]

[23] ¿Qué empresas han marcado tendencia en OSH? [Online] Disponible en: http://www.openhacks.com/page/novedades/id/6/title/%C2%BFQu%C3%A9-es-el-hardware-libre,-la-nueva-%E2%80%9Cmovida%E2%80%9D-tecnol%C3%B3gica-que-llega-a-la-Argentina [Consultado: Junio 18, 2014]

[24] El negocio del hardware en open source, [Online] Disponible en: http://www.euskadinnova.net/es/innovacion-social/noticias/negocio-hardware-open-source/6647.aspx [Consultado: Junio 18, 2014]

[25] Codina Barbera, Marc. 5213: Crear dispositivo para personas sordas (plataforma hardware Arduino). Proyecto de fin de carrera de Ingeniería en Informática. Bellaterra. Universidad Autónoma de Barcelona. Escuela Técnica Superior de Ingeniería, 2013. 51 p.

[26] Ó. T. Artero, “Hardware Arduino,” en Arduino: Curso Práctico de Formación, Ed. RC Libros, Madrid, 2013, pp. 61-128.

[27] Arduino, [Online]. Disponible en: http://www.arduino.cc/es/#.UzL7hvl5N1Y [Consultado: Marzo 4, 2014]

[28] Arduino, [Online]. Disponible en: https://es.scribd.com/doc/240130661/7/Figura-2-4-Diagrama-de-la-arquitectura-AVR [Consultado: Junio 2, 2015]

[29] Análisis comparativo de las placas Arduino (oficiales y compatibles), [Online]. Disponible en: http://comohacer.eu/analisis-comparativo-placas-arduino-oficiales-compatibles/ [Consultado: Junio 4, 2015]

[30] Arduino ¿Qué es Arduino? [Online]. Disponible en: https://arquitecturadecomputadora.wordpress.com/2013/06/07/arduino/ [Consultado: Junio 4, 2015]

[31] Pandaboard, [Online]. Disponible en: http://archlinuxarm.org/platforms/armv7/ti/pandaboard [Consultado: Junio 17, 2014]

[32] Pandaboard: is Open OMAP™ 4 mobile software development platform. Pandaboard ES Technical Specs, [Online]. Disponible en: http://pandaboard.org/content/platform [Consultado: Junio 9, 2015]

[33] L. Iacono, P. Godoy, O. Marianetti y C. García Garino. “Estudio de Plataformas de Hardware Empleadas en
Redes de Sensores Inalámbricas”, presentado en: XVI CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN, Cacic, 2010.

[34] M. Chirinos Colugna, “Configuración de una puerta de enlace en Debian GNU/Linux”, Desarrollo Tecnológico. Disponible en: http://aurea-dt.com/documentos/ADT_PuertaDeEnlace_Manual.pdf

[35] Puerta de enlace o Gateway, [Online]. Disponible en: http://www.codigomaestro.com/redes/puerta-de-enlace-o-gateway/ [Consultado: Julio 2,2015]

[36] VELA, Gustavo Martín. Diseño e implementación de laboratorio remoto para la experimentación con el principio de Arquímedes mediante arquitectura asíncrona distribuida. Proyecto fin de grado de Ingeniería en Tecnologías de Telecomunicación. Bilbao. Universidad de Deusto. Facultad de Ingeniería, 2013. 125 p.

[37] Comparativa y análisis: Raspberry Pi vs competencia, [Online]. Disponible en: http://comohacer.eu/comparativa-y-analisis-raspberry-pi-vs-competencia/
[Consultado: Junio 17, 2015]

[38] Placa de desarrollo Intel Galileo Gen 2, [Online]. Disponible en: http://www.intel.la/content/www/xl/es/do-it-yourself/galileo-maker-quark-board.html
[Consultado: Junio 17, 2015]

[39] Nueva placa Gizmo 2, [Online]. Disponible en: http://www.tecnogaming.com/2014/11/nueva-placa-gizmo-2/ [Consultado: Junio 17, 2015]

[40] Arduino Software (IDE), [Online]. Disponible en: https://www.arduino.cc/en/Guide/Environment [Consultado: Junio 17, 2015]

[41] MARTINEZ, David Luis. SISTEMAS OPERATIVOS. Corrientes, Argentina, 2009, 899 p.

[42] SISTEMAS OPERATIVOS EMBEBIDOS, [Online]. Disponible en: https://chsos20122908514.wordpress.com/2012/11/02/sistemas-operativos-embebidos/ [Consultado: Junio 18, 2015]

[43] D.A. Pérez, “Sistemas Embebidos y Sistemas Operativos Embebidos”, Escuela de computación, Universidad Central de Venezuela, Caracas, Octubre 2009.

[44] D. Azcurra, D. Rodríguez, P. Pytel, D. Santos, V. Giordano, H. Arboleya y R. García, “Arquitecturas de sistemas embebidos utilizables en robótica autónoma,” en XIII Workshop de Investigadores en Ciencias de la Computación, Lanús, Argentina, pp. 702-706, Mayo 2011.

[45] Sobre Linux, [Online]. Disponible en: http://www.linux-es.org/sobre_linux [Consultado: Marzo 6, 2014]

[46] A. F. Montoro, “Introducción: ¿Qué es Linux y que es una Distro?,” En Cámbiate a Linux, Ed. RC Libros, Madrid, 2011, pp.1-22.

[47] Sobre Linux, [Online]. Disponible en: http://www.linux-es.org/sobre_linux [Consultado: Junio 18, 2015]

[48] ¿Qué es una librería? [Online]. Disponible en: http://aprenderinternet.about.com/od/Glosario/fl/Que-es-una-libreria.htm [Consultado: Junio 19, 2015]

[49] Desarrollo Móvil Multiplataforma, [Online]. Disponible en: http://desarrollomovilmultiplataforma.blogspot.com.co/2012/08/aspectos-teoricos-libreria-biblioteca.html [Consultado: Junio 19, 2015]

[50] Definición de script, [Online]. Disponible en: http://www.alegsa.com.ar/Dic/script.php [Consultado: Junio 19, 2015]

[51] OLIVARES FLORES, Linda I. Manual de Programación en Lenguaje C++. Proyecto de Investigación: Métodos de Funciones de Base Radial para la Solución de EDP. México, D.F. UNAM, 2008. 33 p.

[52] LAS TECNOLOGÍAS WIFI Y WIMAX, [Online]. Disponible en: http://www.dip-badajoz.es/agenda/tablon/jornadaWIFI/doc/tecnologias_wifi_wmax.pdf [Consultado: Junio 19, 2015]

[53] Dignani, Jorge Pablo. ANÁLISIS DEL PROTOCOLO ZIGBEE. Trabajo final integrador de especialiazación en Redes y Seguridad. La Plata. Universidad Nacional de La Plata. Facultad de Informática, 2011. 39 p.

[54] Moreno, Javier Martín. Ruiz Fernández, Daniel. Informe Técnico: Protocolo ZigBee (IEEE 802.15.4). Junio de 2007

[55] Validation Environment, [Online]. Disponible en: http://pandaboard.org/node/13531/#debug [Consultado: Junio 17, 2014]

[56] Running Yocto on Pandaboard, [Online]. Disponible en: https://maniacbug.wordpress.com/2012/08/03/pandayocto/ [Consultado: Junio 28,2014]

[57] Ångström Manual –Embedded Power-, [Online]. Disponible en: http://www.student.montefiore.ulg.ac.be/~merciadri/angstrom/files/angstrom-manual.pdf [Consultado: Septiembre 18,2014]

[58] Linaro Community Release for Pandaboard, [Online]. Disponible en: http://releases.linaro.org/14.04/android/panda/ [Consultado: Julio 8,2014]

[59] Embedded Linux system development, [Online]. Disponible en: http://free-electrons.com/doc/training/embedded-linux/embedded-linux-slides.pdf [Consultado: Septiembre 18,2015]

[60] Pandaboard, [Online]. Disponible en: https://eewiki.net/display/linuxonarm/PandaBoard#PandaBoard-CopyRootFileSystem [Consultado: Enero 12,2015]

[61] About the Linaro Ubuntu Pandaboard Release, [Online]. Disponible en: http://releases.linaro.org/15.01/ubuntu/panda/ [Consultado: Agosto 18,2015]

[62] About the Linaro Ubuntu Pandaboard Release, [Online]. Disponible en: https://releases.linaro.org/14.04/ubuntu/panda/ [Consultado: Octubre 22,2015]

[63] UUCP cu, [Online]. Disponible en: https://www.safaribooksonline.com/library/view/building-embedded-linux/059600222X/ch04s10.html [Consultado: Noviembre 7,2015]

[64] Connecting Using UUCP, [Online]. Disponible en: http://www.idevelopment.info/data/Unix/Linux/LINUX_UsingSerialConsoles.shtml [Consultado: Noviembre 7,2014]

[65] Definición de un Access Point, [Online]. Disponible en: http://www.informaticamoderna.com/Acces_point.htm [Consultado: Septiembre 18,2014]

[66] hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator, [Online]. Disponible en: https://w1.fi/hostapd/ [Consultado: Noviembre 18,2014]

[67] WL12xx NLCP Build Instructions, [Online]. Disponible en: http://processors.wiki.ti.com/index.php/WL12xx_NLCP_Build_Instructions [Consultado: Septiembre 3,2015]

[68] Raspberry Pi: Creating A WiFi Access Point, [Online]. Disponible en: http://blog.claytonn.com/raspberry-pi-creating-access-point/ [Consultado: Junio 16,2015]

[69] How To Set Up A Wireless Hotspot (Access Point Mode) That Supports Android In Ubuntu, [Online]. Disponible en: http://www.webupd8.org/2013/06/how-to-set-up-wireless-hotspot-access.html [Consultado: Agosto 17,2015]

[70] Que es un compilador,[Online]. Disponible en: http://ingsistemascompilador.blogspot.com.co/p/conceptos-basicos-sobre-compiladores.html [Consultado: Noviembre 30,2015]

[71] Enlazador,[Online]. Disponible en: http://programaciondesistemasenlazadores.blogspot.com.co/[Consultado [Noviembre 30,2015]

[72] Cron & crontab, explicados ,[Online]. Disponible en:http://blog.desdelinux.net/cron-crontab-explicados/

[73] Demonio (informática), [Online]. Disponible en: https://es.wikipedia.org/wiki/Demonio_(inform%C3%A1tica) [Consultado: Noviembre 26,2015]

[74] Crear un servicio en Linux, [Online]. Disponible en: https://geekytheory.com/crear-un-servicio-en-linux/ [Consultado: Mayo 27,2015]

[75] Script para arranque automático en inicio del sistema Linux (init.d), [Online]. Disponible en: http://rm-rf.es/script-arranque-automatico-sistema-linux-init-d/ [Consultado: Septiembre 3,2015]

[76] Lanzar un script al arranque de Raspberry Pi o Linux, [Online]. Disponible en: http://blog.ingenieriaaremi.com/2013/11/lanzar-script-arranque-raspberry-pi-linux/ [Consultado: Septiembre 5,2015]

[77] Cómo ejecutar un programa automáticamente al arrancar la Raspberry Pi, [Online]. Disponible en: http://nideaderedes.urlansoft.com/2013/12/20/como-ejecutar-un-programa-automaticamente-al-arrancar-la-raspberry-pi/ [Consultado: Septiembre 5,2015]

[78] Crear un demonio en Linux - Resumen y ejemplo, [Online]. Disponible en: http://www.prototipando.es/inicio/apuntes/16-create-daemon-on-linux?showall=&start=2 [Consultado: Septiembre 18,2015]

[79] Crear Demonio en Linux (Servicios), [Online]. Disponible en: https://liberatucodigo.wordpress.com/2012/05/19/crear-demonio-en-linux-servicios/ [Consultado: Septiembre 20,2015]

[80] Como funcionan los runlevel del sistema, [Online]. Disponible en: https://arenlasysadmin.wordpress.com/2013/05/05/configurar-runlevel/ [Consultado: Septiembre 21,2015]

[81] Controlando los servicios y aplicaciones al inicio en GNU/Linux, [Online]. Disponible en: http://fraterneo.blogspot.com.co/2011/01/controlando-los-servicios-y.html [Consultado: Septiembre 21,2015]

[82] Arduino porting to Atmega48, [Online]. Disponible en: http://www.thinkcreate.org/index.php/arduino-porting-to-atmega48/ [Consultado: Mayo 18,2015]

[83] Embedded Programming, [Online]. Disponible en: http://fablabuni.edu.pe/embedded-programming/ [Consultado: Mayo 19,2015]

[84] WiFi library, [Online]. Disponible en: https://www.arduino.cc/en/Reference/WiFi [Consultado: Noviembre 30,2015]

[85] SPI library, [Online]. Disponible en: https://www.arduino.cc/en/Reference/SPI [Consultado: Noviembre 30,2015]

[86] ScanNetworks.ino, [Online]. Disponible en: https://github.com/codebendercc/arduino-library-files/blob/master/libraries/WiFi/examples/ScanNetworks/ScanNetworks.ino [Consultado: Noviembre 30,2015]

[87] Send and Receive UDP String, [Online]. Disponible en: https://www.arduino.cc/en/Tutorial/WiFiSendReceiveUDPString [Consultado: Noviembre 30,2015]

[88] Protocolo SSH, [Online]. Disponible en: http://www.gb.nrao.edu/pubcomputing/redhatELWS4/RH-DOCS/rhel-rg-es-4/ch-ssh.html [Consultado: Noviembre 25,2015]

[89] The Network File System, [Online]. Disponible en: http://www.tldp.org/LDP/nag/node140.html [Consultado: Noviembre 25,2015]

[90] Compartir archivos mediante NFS (Network File System), [Online]. Disponible en: http://www.kubuntu-es.org/wiki/internet-redes/compartir-archivos-mediante-nfs-network-file-system [Consultado: Noviembre 25,2015]

[91] Carlos Enrique Serrano, Modelo integral para el profesional en ingeniería. Popayán, Cauca, 2005.

[92] S. Castaño, Un Modelo Integral para un Profesional en Ingeniería. Popayán, Cauca, 2003.