UNIVERSIDAD DEL AZUAY
FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
ESCUELA DE INGENIERÍA DE SISTEMAS

Tema:

Clúster de servidores mediante dispositivos RaspberryPi2

Trabajo de titulación previo a la obtención del título de

Ingenieros en Sistemas

Autores:

Fausto Sarango Molina

Rubén Riofrio Ocaña

Director:

Ing. Marcos Orellana Cordero

CUENCA-ECUADOR

2016

DEDICATORIA

El presente trabajo va dedicado principalmente a Dios, por iluminarme y estar de mi lado en todo momento y a lo largo de mi carrera.

A mi madre, mi familia por la ayuda desinteresada que me brindaron en cada obstáculo que la vida me presento, gracias a sus ejemplos, consejos, que siempre fueron encaminados hacia mi superación personal.

A mi hija que es la forjadora e inspiradora de todos mis logros y metas. A mis amigos por su apoyo y ayuda incondicional.

Fausto Vinicio Sarango Molina

DEDICATORIA

Dedico este tema de investigación a Dios, quien siempre ha sido mi guía e inspiración para alcanzar todas la etapas de mi vida, gracias señor por todas la bendiciones recibidas.

A mi madre quien siempre me apoyado incondicionalmente en el desarrollo de mi vida y en todas mis decisiones. A mi esposa Fer que desde que entro en mi vida, ha sido un pilar fundamental para alcanzar mis metas y logros. A mis hijos Adrián y Sarita quienes han sido mi fuente de inspiración y mi fortaleza para seguir adelante día tras día.

A todos mis amigos y familiares porque con ellos empezamos este camino y siempre estuvieron presentes para apoyarnos y brindarnos su ayuda incondicional.

Ruben Omar Riofrio Ocaña.

AGRADECIMIENTOS

A mi madre y familia, que a lo largo de toda mi vida me han apoyado y motivado para seguir adelante con mis estudios, creyendo en mí y en que la educación es la mejor herencia que se le puede dejar a un ser querido.

A nuestro director de trabajo de titulación Ing. Marcos Orellana Cordero, gracias por el apoyo brindado a lo largo de todo el proceso de desarrollo del presente trabajo.

Fausto Vinicio Sarango Molina

AGRADECIMIENTOS

Quiero dar Gracias a Dios, por la oportunidad de terminar esta etapa de mi vida y ser la fortaleza para seguir siempre adelante. A mi madre quien me apoyo y me dio su guía para seguir adelante.

A mi esposa y mis hijos quienes siempre me apoyaron y fueron mi inspiración para culminar esta etapa de mi vida. A los docentes de la universidad del y a la escuela de sistemas, por habernos impartido los conocimientos necesarios para culminar esta etapa de nuestras vidas.

Agradezco a nuestro director de tesis Ing. Marcos Orellana Cordero, quien fue el pilar fundamental para el desarrollo de esta tesis, quien nos guio y aconsejo para la finalización de este tema de investigación.

Ruben Omar Riofrio Ocaña.

Contenido

DEDICATORIA **2**
AGRADECIMIENTOS **4**
Resumen **8**
ABSTRACT **9**
Introducción **10**
Objetivos **11**
Objetivo general 11
Objetivos específicos 11
Capítulo 1: Sistema Cliente-Servidor **11**
1.1 Servicios cliente-servidor 12
1.1.1 Servidor de archivos 13
1.1.2 Servidor de impresión 13
1.1.3 Servidor de aplicaciones 13
1.1.4 Servidor de correo 14
1.1.5 Servidor de fax 14
1.1.6 Servidor de Servicios de Directorio 14
1.1.7 Servidor web 14
1.1.8 Servidor de Base de Datos 15
1.1.9 Servidores Transaccionales 15
1.2 Funciones cliente-servidor 15
1.2.1 Las principales operaciones del servidor se enumeran a continuación: 16
1.2.2 Las principales operaciones del cliente se enumeran a continuación: 16
1.3 Clúster 16
1.3.1 Características de los Clúster 17
1.4 Seguridad informática 18
1.4.1 Principales objetivos de la seguridad informática 19
1.5 Alta disponibilidad 21
1.6 Computación Paralela 22
1.6.1 Que es la computación paralela? 23
1.6.2 Computación en serie: 24
1.6.3 Computación Paralela: 24
1.7 Sistemas Distribuidos. 26

CAPITULO 2: Tecnologías Aplicadas..... 29
2.1 Introducción 29
2.2 Dispositivo Raspberrypi2 30
2.3 Procesadores ARMv7 35
2.3.1Características Básicas ARM 37
CAPITULO 3: Desarrollo e implementación..... 40
3.1 Sistemas operativos que soporta RaspberryPi y características del dispositivo. 40
3.1.1 Instalación del Sistema Operativo 43
3.2 Clúster con RaspberryPi2 55
3.2.1 Clúster servidor RaspberryPi2 55
3.2.2 Clúster servidor RaspberryPi2, características y esquema 57
3.3 Configuración de la Red 59
3.3.1 Configuración de la Red del Nodo Maestro 59
3.3.2 Configuración de la Red del Nodo Esclavos 63
3.4 Conectar una USB en el nodo maestro 66
3.4.1 Configuración de los nodo maestro para que el disco-USB se pueda replicar en cada uno de los nodos que forman parte del clúster. 66
3.4.2 Configuración de los nodos esclavos para que el disco-USB se pueda replicar en ellos 70
3.5 Configuración del Servidor FTP 75
3.5.1 Instalación del Servidor FTP 75
3.5.2 Creación de Carpetas Compartidas 76
3.5.3 Acceso a los Usuarios 77
3.6 MySQL 82
3.6.1 Instalar MySQL 82
3.6.2 Mover carpeta MySQL en la Unidad externa 85
3.7 Alta Disponibilidad 86
3.7.1 Instalación del Pacemaker y Cronosync 87
3.7.2 Configuración de la IP Virtual 89
3.7.3 Levantamiento de servicios Pacemaker y Cronosync 90
3.7.4 Configurar Pacemaker 90
Conclusiones..... 96
Glosario 98
Bibliografía..... 99

Resumen

Este estudio realiza una investigación sobre el dispositivo RaspberryPi2 y su tecnología conexa. Para el efecto, se simula un sistema cliente-servidor con características similares a las de un servidor tradicional. Se detalla paso a paso la implementación de hardware y software que se utilizó para la construcción y puesta a punto del clúster de servidores y la instalación de los servicios relacionados. Así mismo, se demuestra la vialidad del prototipo con pruebas de rendimiento y alta disponibilidad. La disponibilidad de este sistema contribuiría a que las pequeñas y medianas empresas puedan desarrollar servicios que anteriormente estaban restringidos por su costo y tamaño a grandes organizaciones.

ABSTRACT

This study deals with a research on the Raspberry Pi 2 device and its related technology. For this purpose, a client-server system with similar features to those of a traditional server is simulated. The implementation of hardware and software used for the construction and tuning of the server cluster as well as the installation of related services is detailed step by step. Likewise, the viability of the prototype with testing performance and high availability is demonstrated. The availability of this system would help small and medium-sized enterprises to develop services that were previously restricted only to large organizations due to its cost and size.

UNIVERSIDAD DEL AZUAY
Dpto. Idiomas

Translated by,
Lic. Lourdes Crespo

Introducción

La investigación de nuevos dispositivos que ayuden a suplir las necesidades empresariales y que garanticen mayor rapidez con menores costos de implementación en el área de producción, han sufrido cambios vertiginosos en nuestro medio, por lo cual obliga a estar inmersos en el mundo de la búsqueda de nuevos dispositivos. Es por ello que, el enfoque de este proyecto está dirigido hacia un dispositivo que está siendo utilizado en los países de primer mundo llamado *RaspberryPi2*, que podría también ser implementado en el Ecuador y apoyar al crecimiento de las empresas pymes.

La mayoría de las grandes empresas que tienen un continuo desarrollo, cuentan con servidores de altas prestaciones, pero también de muy altos costos, es por ello que hemos revisado las necesidades de las pequeñas empresas que requieran de este servicio, un sistema cliente-servidor que supla sus necesidades y que los ayude a crecer, adaptable a la realidad empresarial. Es indispensable investigar nuevas tecnologías que ayuden al sector MPYMES a suplir estos requerimientos. Otro campo que se abordará y que es de vital importancia, será brindar un respaldo en el procesamiento de información, que pueda ser visualizado desde cualquier parte del mundo.

El sistema *RaspberryPi2* ya está siendo utilizado en países del primer mundo como; EE.UU, Inglaterra, y parte de Europa, lo que ha producido un desarrollo de nuevas implementaciones de este dispositivo en distintas áreas tecnológicas y puntualmente en lo que a servidores se refiere que es el tema de investigación. Se ha hecho muchas innovaciones, obteniendo muy buenos resultados y mejorando los servicios que prestan los servidores tradicionales, que es el objetivo de todo tema de investigación.

En este mundo dominado por la tecnología y por las ventas que se realizan por medio del internet, es necesario que para el desarrollo empresarial se cuente con servidores de muy altas prestaciones que brinden seguridad y estén propensos a la adaptabilidad, dependiendo del crecimiento de las Empresas. Es por ello que la incesante investigación da como resultado la creación de dispositivos mucho más adaptables a este mundo tecnológico que brinden mejores servicios que otros tradicionales y que su costo sea muy económico, como *RaspberryPi2.*

Las tecnologías y dispositivos que hoy se utilizan, son resultados de la continua investigación para incorporarlos al medio de producción empresarial, estudiando las características y funcionalidades con las que cuentan, permitiendo crear nuevas áreas de implementación en el mundo.

Objetivos

Objetivo general

Implementar un Sistema Cliente-Servidor mediante tecnología *RaspberryPI2* que garantice los mismos beneficios de un servidor tradicional, de alto costo, a un costo asequible para Empresas e Instituciones de tipo pymes.

Objetivos específicos

1. Sistematizar conceptos y fundamentos de la tecnología *RaspberryPI2*.
2. Desarrollar un sistema cliente–servidor con *RaspberryPI2*.
3. Implementar servicios de redes y base de datos en el clúster
4. Generar pruebas de rendimiento y disponibilidad

Capítulo 1: Sistema Cliente-Servidor

De acuerdo a los SIG (Sistemas de Información Gerencial), la computación Cliente-Servidor es una tecnología del modelo de aplicación distribuida, que permite brindar soluciones a grandes empresas, en lo que se refiere a problemas de gestión de datos, en donde las tareas se reparten entre los proveedores de recursos o servicios llamados servidores, y las demandas llamadas clientes. Este modelo basa sus funciones entre los dos tipos de procesos independientes. El cliente es cualquier proceso, que hace una petición al servidor. Mientras que un servidor es aquel que emite la respuesta requerida

por el cliente. Este modelo también puede aplicarse sobre un solo computador o en varios equipos conectados en red. Cuando este servicio reside sobre ordenadores independientes en una red, el servidor puede dar servicio a más de un cliente, es decir, el cliente puede pedir información a varios servidores, independientemente de la ubicación física de los mismos. La red que vincula tanto al cliente como al servidor como un conjunto, proporciona un medio de comunicación entre ellos. (Yadav, 2009)

Modelo básico de computación cliente-servidor

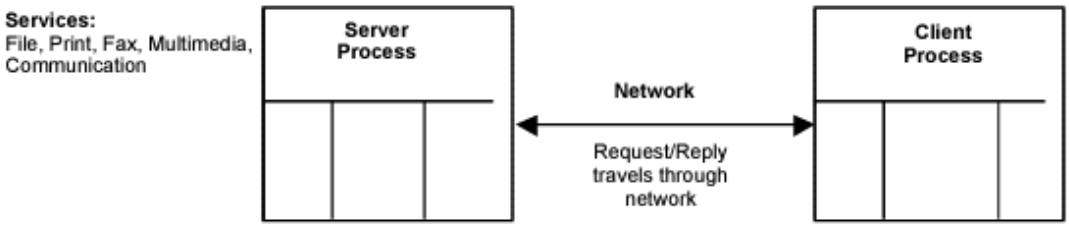


Fig. 1: Modelo básico de computación cliente-servidor (Yadav, 2009).

En la figura se muestra como los servicios pueden ser proporcionados por cualquiera de los equipos de la red. Lo más importante del modelo cliente-servidor es donde se lleva a cabo el procesamiento de la solicitud. Es decir, en el Sistema de base de datos del cliente-servidor su funcionalidad se divide entre el servidor y varios clientes. (Yadav, 2009)

1.1 Servicios cliente-servidor

Los servidores de datos pueden almacenar desde información muy simple hasta información muy compleja, es decir, cualquier tipo de datos y para ello, solamente es necesario un servidor. En sistemas más grandes y con mayor complejidad, la responsabilidad puede ser distribuida en los diferentes tipos de servidores. A continuación, se explicará algunos propósitos de varios servidores, que brindan varios servicios: (Yadav, 2009)

1.1.1 Servidor de archivos

Todos y cada uno de los archivos se encuentran en una máquina servidor, este servidor proporciona a los clientes acceso a los registros dentro de los archivos de la máquina servidor, el proceso es útil para intercambiar archivos a través de la red, ya que los clientes pueden solicitar diferentes servicios. Cabe recalcar que este proceso es antiguo, por lo que deben existir varios intercambios de mensajes por la red, para encontrar los datos que el cliente ha solicitado. (Yadav, 2009)

Ejemplos de servidores de archivos:

- UNIX: NFS (Network File Services) creados por los sistemas Sun Microsystem
- Microsoft Windows "Map Drive" e.g., Rivier College's "P-drive".
- Samba: Un Source/Free de software libre de código abierto que ofrece servicios de archivos sin fisuras y de impresión a clientes SMB / CIFS (i.e., Microsoft Windows clients). (Yadav, 2009)

1.1.2 Servidor de impresión

Este servidor permite configurar una máquina para tener acceso a los periféricos de salida compartidos; el más común es la impresora, estos servidores fueron los primeros en ser configurados en las empresas grandes, ya que era muy costoso tener una impresora con altas características para cada departamento. Los servicios de impresión se pueden ejecutar en un servidor de archivos o en una o varias máquinas del servidor de impresión separadas. (Yadav, 2009)

1.1.3 Servidor de aplicaciones

Este tipo de configuración de servidores es utilizada para gestionar el acceso al software de aplicaciones centralizadas. Lo que significa que las aplicaciones son instaladas en el servidor y los clientes solo pueden acceder a la información requerida; ellos no pueden ver qué proceso se está corriendo para obtener dicha información. (Yadav, 2009)

1.1.4 Servidor de correo

Este servicio es configurado para manejar el flujo de correo electrónico, mensajería y comunicación con los sistemas mainframe (Mainframe es un gran ordenador capaz de realizar el procesamiento de datos complejos.) en redes de gran escala. (Yadav, 2009)

1.1.5 Servidor de fax

Esta es una configuración adicional, que se hace para enviar y recibir los faxes a través de una única conexión de red. Esta configuración se la puede realizar, utilizando un tablero de fax que servirá como una estación de trabajo única, que requiere de un software o un dispositivo dedicado para este servicio. Esta configuración es muy similar al servidor de correo.

1.1.6 Servidor de Servicios de Directorio

Esta configuración se la hace cuando se tiene sistemas de clientes a gran escala distribuidos a lo largo de varios servidores, y que cada servidor puede estar en distintos lugares físicos, pero que abarcan cada uno de ellos grandes cantidades de información. Este servicio funciona como un administrador de la organización, ya que hace un seguimiento de lo que se almacena en él, lo que brindará un acceso más rápido y confiable de los datos requeridos por los clientes independientemente del lugar donde se realice la petición. (Yadav, 2009)

1.1.7 Servidor web

Esta máquina es configurada para almacenar y recuperar datos de Internet e Intranet de la organización: datos de consulta, documentos y otros que residen en él. Esta aplicación web proporcionará todo el acceso y las restricciones a los distintos documentos, tales documentos se comparten a través de una Internet, Intranet, Extranet. El protocolo estándar para el desarrollo de servidores de aplicaciones Web es HTTP (Hyper Text Transfer Protocol). (Yadav, 2009)

1.1.8 Servidor de Base de Datos

El Servidor de BD proporciona acceso a los datos, a partir de peticiones realizadas por los clientes, este servidor almacena los datos, en forma de una BD SQL (*Structured Query Language*), es decir comparte los datos que residen en la BD en una red, este utiliza un protocolo más eficiente que el servidor de archivos. El servidor de BD recibe las solicitudes SQL de los clientes, los procesa, y envía solo los datos solicitados, por lo tanto el cliente no recibe información no deseada o irrelevante, la única restricción es que el cliente debe aplicar el código SQL para sus peticiones. Un servidor de BD es el: Oracle9i. (Yadav, 2009)

1.1.9 Servidores Transaccionales

En este tipo de servidores los datos y procedimientos residen en un servidor, a su vez este debe seguir un protocolo, que brinda acceso a funciones de alto nivel, y aplica un procedimiento transaccional muy eficiente, generalmente estos servidores utilizan OLTP (**O**n**L**ine **T**ransaction **P**rocessing) para implementar el procesamiento transaccional en línea, en aplicaciones de alto rendimiento. Este servicio utiliza un protocolo con mayor eficiencia que el utilizado en los servidores de BD. A continuación se dan a conocer algunos servidores transaccionales.

- TP-Light con la base de datos de procedimientos almacenados como Oracle, Microsoft SQL Server, etc.
- TP-Heavy con monitores TP como BEA Tuxedo, IBM CICS / TX Series. (Yadav, 2009)

1.2 Funciones cliente-servidor

Las principales funciones del cliente-servidor se enumeran a continuación:

- Manejo de la interfaz de usuario.
- Captura y valida la sintaxis de los datos de entrada proporcionados por el usuario.
- Gestiona los procesos lógicos de la aplicación.
- Generación de Consultas e informes sobre la BD.

- Genera una respuesta de vuelta al servidor.

1.2.1 Las principales operaciones del servidor se enumeran a continuación:

- Acepta los requerimientos de BD que hacen los clientes.
- Valida las autorizaciones.
- Asegura que las restricciones de integridad no sean violadas.
- Realiza el procesamiento de consulta/actualización a nivel de BD y transmite las repuestas al cliente.
- Mantiene un catálogo del sistema.
- Facilita el acceso de BD concurrentes.
- Suministra un control de Recuperación. (Yadav, 2009)

1.2.2 Las principales operaciones del cliente se enumeran a continuación:

- Interactúa con el usuario.
- Administra la interfaz del usuario.
- Genera los requerimientos de BD.
- Realiza validaciones locales y procesa la lógica de la aplicación.
- Recibe los resultados del servidor.
- Formatea los resultados. (Yadav, 2009)

1.3 Clúster

Desde que se crearon las redes informáticas, hubo un interés de crear comunicación entre varias computadoras básicas, de ahí el denominado termino clúster, por la agrupación o unión de varias computadoras, en la década de los 60 la empresa ARPANET contaba con cuatro tipos de centros informáticos, en donde cada uno de ellos contaba con varias máquinas que estaban conectadas entres si, como un grupo, es por ello el nombre de clúster informático. (Yadav, 2009)

Hoy en día con el surgimiento de nuevos dispositivos, avance tecnológico y con la investigación de nuevos usuarios sobre los avances informáticos, surgen nuevos requerimientos, y con ello la necesidad de ampliar el concepto de clúster de computadoras, que dejó de ser un mito investigativo y se transformó en un tema de

actualidad, que da cabida a nuevas indagaciones que mejoran la calidad de servicio de los servidores Web, garantizando y asegurando que los recursos empresariales están siempre seguros y disponibles. (Yadav, 2009)

Un clúster informático, es un grupo de sistemas independientes, conocidos como nodos, que trabajan de manera conjunta como uno solo, garantizando que la carga de trabajo sea balanceada, obteniendo un alto rendimiento, y una disponibilidad de los datos requeridos por el usuario. (Yadav, 2009)

1.3.1 Características de los Clúster

Cuando se crea un clúster se espera que garantice la combinación óptima de 4 características esenciales: alto rendimiento, alta disponibilidad, balanceo de carga y escalabilidad. Además, cuando se crea un clúster de computadoras es más fácil y económico por la utilización de la misma configuración en la mayoría de equipos. Esto conlleva a un ahorro económico, por la utilización del mismo sistema operativo y de hardware.

- Alto rendimiento.- son clústeres creados para ejecutar tareas que requieren de una gran capacidad de procesamiento, además requieren también grandes cantidades de memoria o inclusive ambas a la vez. La utilización de este tipo de clúster es requerido porque los problemas a resolver involucran grandes cantidades de recursos por largos periodos de tiempo.
- Alta disponibilidad.- La creación de estos clústeres, tienen la finalidad de garantizar al cliente: disponibilidad y confiabilidad, además de mantener varios servicios compartidos, que están constantemente respaldándose entre sí. La confiabilidad se la da el software que detecta fallos y permite recuperase frente a estos, mientras que el hardware evita tener un único punto de fallo.
- Balanceo de carga.- esto hace referencia a compartir el trabajo que se va a realizar entre varios procesos, ordenadores, discos duros u otros recursos; esto se mantiene gracias a un algoritmo, que se utiliza para dividir de la forma más equilibrada, la tarea en proceso de ejecución, y así evitar los cuellos de botella.
- Escalabilidad.- Esta es una propiedad deseable por todo sistema, red o proceso, que indica su robustez frente a cambios bruscos por diferentes circunstancias y

que garantiza su misma calidad de los servicios ofrecidos pese a dichos inconvenientes.

Cabe recalcar que, para que un clúster funcione con estas 4 características, no solo basta con conectar los computadores entres si, al contrario, en la actualidad existen varios sistemas de manejos de clústeres, los cuales se encargan de interactuar tanto con el usuario y los procesos que se ejecutan internamente sobre estos, para así lograr una mayor optimización en su funcionamiento. (Yadav, 2009)

1.4 Seguridad informática

Las TIC (Tecnologías de Información y la Comunicación) y en especial la informática, han desarrollado una área específica para el tratamiento de la seguridad, llamada seguridad informática, que se enfoca en brindar protección a la infraestructura computacional y todo lo relacionado con sistemas de computación, y en especial todo lo que en ellos contienen, que son; datos, información circulante y algunos otros recursos.

Los sistemas cliente – servidor, no están exentos de la seguridad de la información, ya que su principal función es ofrecer a los clientes la posibilidad de compartir datos, información y recursos de hardware y software. La ventaja de montar un servidor en computadoras dedicadas a este fin, es la seguridad. Por esta razón la mayoría de los servidores son procesos diseñados, de forma que puedan funcionar en computadoras de propósito específico. (Costas Santos, 2014)

En la actualidad el mundo está globalizado gracias a la utilización del internet, es por ello la expansión de lo que hoy en día se llama cyber seguridad, que busca proteger y brindar seguridad a los datos, tomando en cuenta que el funcionamiento correcto de los sistemas depende en gran medida, de protegerlos como uno de los más grandes activos que existen. (Costas Santos, 2014)

El principal objetivo de la seguridad informática, es asegurar los recursos del sistema informático de cualquier organización, esto se puede lograr brindando niveles de accesibilidad y permitiendo que solo personas autorizadas puedan acceder a realizar ciertas modificaciones dentro del sistema, o permitir la extracción de cierta información autorizada por la organización.

1.4.1 Principales objetivos de la seguridad informática

Existen muchas discrepancias en la elaboración de los objetivos de la seguridad Informática, ya que cada autor de los libros tienen sus propias experiencia en ello. Pero en algunos de estos objetivos existe similitud y se pondrá a consideración.

- Detectar los posibles problemas y amenazas a la seguridad, minimizando y gestionando los riesgos.
- Garantizar la adecuada utilización de los recursos y de las aplicaciones de los sistemas.
- Limitar las pérdidas y conseguir la adecuada recuperación del sistema en caso de un incidente de seguridad.
- Cumplir con el marco legal y con los requisitos impuestos a nivel organizativo.

La seguridad es un tema de constante investigación, ya que el elemento riesgo no se lo puede eliminar por completo, es decir siempre está y estará presente en cualquier aplicación y hasta en el convivir cotidiano, independientemente de las medidas que se tomen, es por ello, que la seguridad ha sido dividida por niveles, cada uno de estos niveles utilizan un conjunto de técnicas encaminadas a obtener altos niveles de seguridad en los sistemas informáticos. Se podría decir que la seguridad es una característica de cualquier sistema que indica que está libre de peligro. Pero en los sistemas informáticos, sistemas operativos o redes de computadoras, es muy difícil de conseguir, es por ello que se habla de los niveles de seguridad como son:

Fiabilidad.- La fiabilidad es la probabilidad de que un sistema se comporte de la manera que se espera, es decir que se habla de contar con sistemas fiables en lugar de seguros. Esta fiabilidad podría tener un tiempo de duración, ya que esta condición puede cambiar por el paso de los años. (Costas Santos, 2014)

Confidencialidad.- Es la propiedad de la información, mensaje, comunicación o datos, con lo que se garantiza que está accesible por la persona o sistema que esté autorizado para acceder a dicha información. (Costas Santos, 2014)

Integridad.- Esta es una cualidad que garantiza que no ha existido modificación alguna en la información, mensaje, comunicación o datos originales. Esta alteración se presenta cuando una persona, programa o procesa, borra o modifica parte de la información que

no ha sido autorizada para dicha actualización o borrado, la integridad se obtiene adjuntando otro conjunto de datos de comprobación, que garanticen que estos datos no han sido alterados. Por ejemplo una firma digital. (Costas Santos, 2014)

Disponibilidad.- Es la cualidad que tiene un servicio, datos, sistema, de encontrarse a disposición de quienes deben acceder a ellos, ya sean personas, procesos o aplicaciones. Esta disponibilidad supone que la información puede ser accesible en el momento que se la requiera, evitando su pérdida o bloqueo. (Costas Santos, 2014)

Autenticación.- Aquí se verifica que un documento, información, o datos, pertenezca a quien está intentando acceder. Aquí se debe crear un sistema de registro de datos, en donde el usuario pueda ser verificado mediante un login y una contraseña o password. (Costas Santos, 2014)

No repudio o irrenunciabilidad.- estrechamente relacionado con la autenticación y permite probar la participación de las partes en una comunicación. Existen dos posibilidades: (Costas Santos, 2014)

- No repudio en origen: el emisor no puede negar el envío. La prueba la crea el propio emisor y la recibe el destinatario.
- No repudio en destino: el receptor no puede negar que recibió el mensaje porque el emisor tiene pruebas de la recepción. En este caso la prueba irrefutable la crea el receptor y la recibe el emisor.

A estas características se las conoce con el nombre de CIDAN, este nombre ha sido puesto con la utilización de la inicial de cada característica, y existe una figura que denota la relación entre estas características como se muestra en la figura.

Fig. 2: Niveles de Seguridad (Costas Santos, 2014).

1.5 Alta disponibilidad

La alta disponibilidad (High Availability), hace referencia a la capacidad de las aplicaciones, datos, para que se encuentren disponibles y listos para ser utilizados por los usuarios autorizados que requieran de ellos, en todo momento y sin ninguna interrupción o problema. El objetivo de la alta disponibilidad es mantener los sistemas corporativos funcionando los 365 días del año. Las interrupciones que podrían suscitarse, pueden ser de dos tipos: (Costas Santos, 2014)

- **Las interrupciones previstas.-** Se presentan cuando los sistemas o procesos son paralizados por un administrador, esto generalmente ocurre cuando se necesita realizar algún mantenimiento, cambios, o mejoras en el hardware o software. (Costas Santos, 2014)
- **Las interrupciones imprevistas.-** Se dice que una interrupción imprevista puede ser ocasionada por un apagón, un error inesperado ya sea de hardware o software, problemas de seguridad o un desastre natural, virus, accidentes, caídas involuntarias del sistema y otras. Dichas interrupciones no se las puede eliminar, pero si estar preparados con otras opciones en el caso de que alguna de estas ocurra. (Costas Santos, 2014)

Las métricas comúnmente utilizadas para medir la disponibilidad y fiabilidad de un sistema son el tiempo medio entre fallos o MTTF (Mean Time To Failure) que mide el tiempo medio transcurrido hasta que un dispositivo falla, y el tiempo medio de recuperación o MTTR (Mean Time To Recover) mide el tiempo medio tomado en restablecerse la situación normal una vez que se ha producido el fallo. El tiempo en el que un sistema está fuera de servicio se mide a menudo como el cociente MTTR/MTTF. Lógicamente, el principal objetivo es aumentar el MTTF y reducir el MTTR de forma que minimice el tiempo de no disponibilidad del servicio. (Costas Santos, 2014)

Existen distintos niveles de disponibilidad del sistema, según el tiempo aproximado de inactividad por año se determina el porcentaje de disponibilidad. El mayor nivel de exigencia de alta disponibilidad acepta 5 minutos de inactividad al año, con lo que se obtiene una disponibilidad de 5 nueves: 99,999%. (Costas Santos, 2014)

Como ejemplos de sistemas y servicios de alta disponibilidad se puede mencionar sistemas sanitarios, control aéreo, de comercio electrónico, bancarios, transporte

marítimo, militares, etc., donde la pérdida o interrupción de conectividad pueden suponer graves consecuencias personales y económicas. En el Capítulo 3 se profundizará en algunas de las técnicas que permiten mejorar la disponibilidad de los sistemas y servicios ofrecidos por estos. (Costas Santos, 2014)

La tecnología cliente-servidor, junto con la computación paralela y la seguridad de la información, son tecnologías de desarrollo que van de la mano una de la otra, brindándose respaldo y apoyo en el desarrollo de clúster de servidores, esto ha conllevado a idear una nueva rama llamada tecnología cliente-servidor BOINC, que consiste en crear dos capaces de operar bajo la arquitectura cliente-servidor, es decir el software es instalado en el computador, el servidor empieza a enviar tareas al cliente, y aquí es donde interviene tanto la seguridad informática como la computación paralela. Todas las operaciones son ejecutadas en el lado del cliente y los resultados son subidos al servidor, su diseño se centró en los desarrolladores que deseen crear un proyecto de computación distribuida, es por ello su utilización en la creación de clúster con dispositivos como RaspberryPI. Por las limitaciones de la placa es necesario el estudio de estos y otras tecnologías aplicadas a este dispositivo. (Costas Santos, 2014)

En la creación de un clúster con el dispositivo RaspberryPI es necesario tener en cuenta todos los beneficios y qué arquitectura es más viable y adaptable a este dispositivo, es por ello que el estudio de la arquitectura cliente-servidor da la posibilidad de distribuir procesos y el trabajo de una manera más eficiente y confiable, así mismo brinda estabilidad a los sistemas informáticos. (Costas Santos, 2014)

Cada proceso puede funcionar en una placa de RaspberryPI como un proceso independiente, ya que para la creación de un clúster es necesario la utilización de varios dispositivos RaspberryPI, y cada uno de ellos puede tener una función específica, para luego en conjunto lograr el objetivo deseado que es la creación de un clúster con RaspberryPI2. (Costas Santos, 2014)

1.6 Computación Paralela

El problema de la computación antigua era cómo se ejecutaban varias instrucciones simultáneamente, esto se daba por la baja capacidad de los equipos o por

desconocimiento de técnicas para mejorar el rendimiento y la ejecución de instrucciones dadas a un computador. La computación paralela es una forma de cómputo en la que varias instrucciones se ejecutan de una forma simultánea, operando sobre los principios que un problema grande se lo divide en problemas pequeños, o simplemente el conocido "divide y vencerás". Esto a menudo se los puede dividir y encontrar una solución para cada uno de estos problemas más pequeños, que luego se resolverá simultáneamente en paralelo, de ahí el nombre de computación paralela.

Estos procesos son realizados mediante MPI (Message Passing Interface), que es un protocolo de comunicación que ofrece una interfaz amigable con el usuario y a su vez estándar para ejecutar aplicaciones de computación paralela. En el libro de Dennis Andrew cita dos implementaciones de esta tecnología: OpenMPI y MPICH. (Dennis, Raspberry Pi Super Cluster , 2013)

MPICH (Message Passing Interface Chameleon). Esta aplicación se basa en una versión de MPI estándar que permite utilizar aplicaciones previamente diseñadas en los lenguajes C, C++ y Fortran. Para realizar esta implementación se utilizará una placa RaspberryPi, que deberá trabajar como una sola entidad, es decir, se implementará dicha aplicación y se realizará el test de prueba.

1.6.1 Que es la computación paralela?

La computación paralela es una forma de cómputo en la que varias instrucciones se ejecutan de una forma simultánea, operando sobre los principios que un problema grande se lo divide en problemas pequeños, o simplemente el conocido "divide y vencerás". (Joubert, 2002)

En la demostración que se hace a continuación, se indica las características y gráficas, tanto de la computación en serie, como la computación paralela. Poniendo en evidencia claramente sus ventajas y desventajas a la hora de ponerlas en práctica, en el desarrollo de los clústers.

1.6.2 Computación en serie:

- Tradicionalmente, el software ha sido escrito para el cálculo en serie:
  - Un problema se divide en una serie discreta de instrucciones.
  - Las instrucciones se ejecutan secuencialmente una tras otra.
  - Ejecutada en un único procesador.
  - Sólo una instrucción puede ejecutar en cualquier momento en el tiempo.

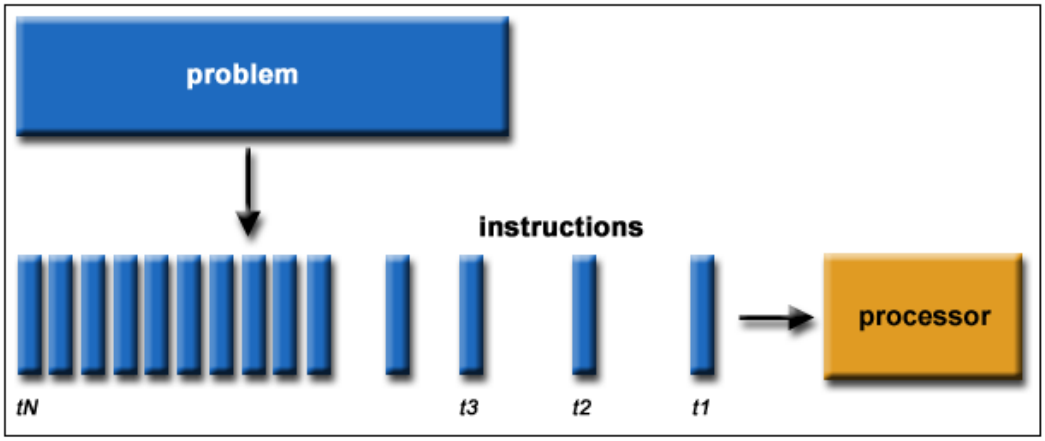


Fig.3: Computación en serie (Blaise Barney, 2015)

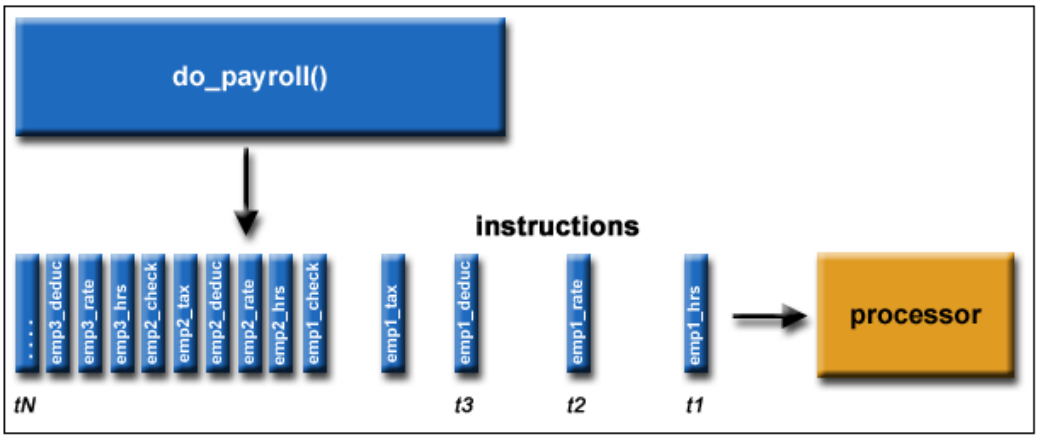


Fig.4: Computación en serie (Blaise Barney, 2015).

1.6.3 Computación Paralela:

- La computación paralela utiliza simultáneamente múltiples recursos informáticos para resolver un problema:

- El problema se divide en partes más pequeñas y así poderlas resolver al mismo tiempo
- A su vez cada parte se divide en una serie de instrucciones
- Estas instrucciones se ejecutan simultáneamente en diferentes procesadores
- Se emplea un mecanismo global de control / coordinación

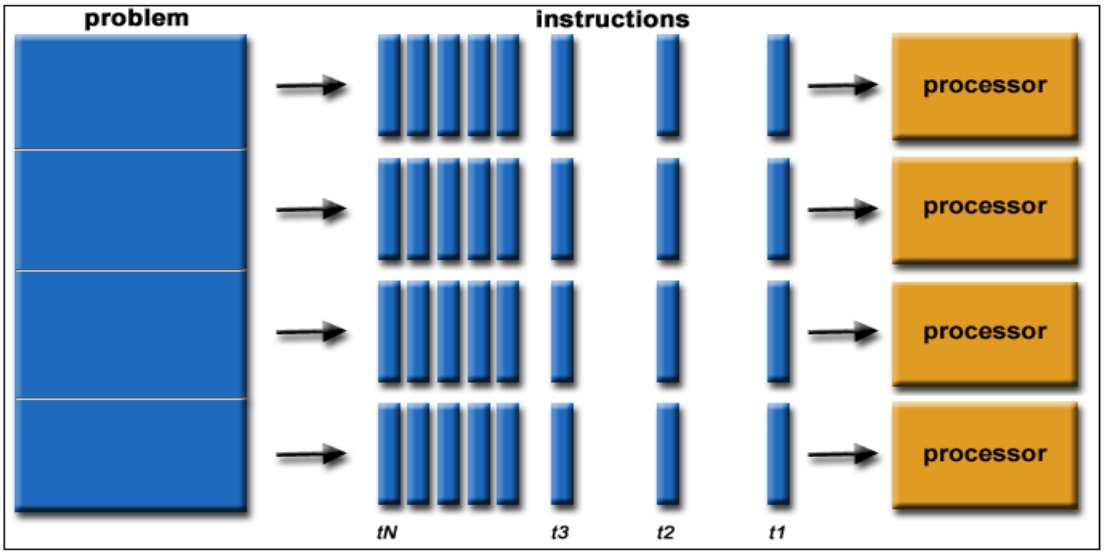


Fig.5: Computación paralela (Blaise Barney, 2015).

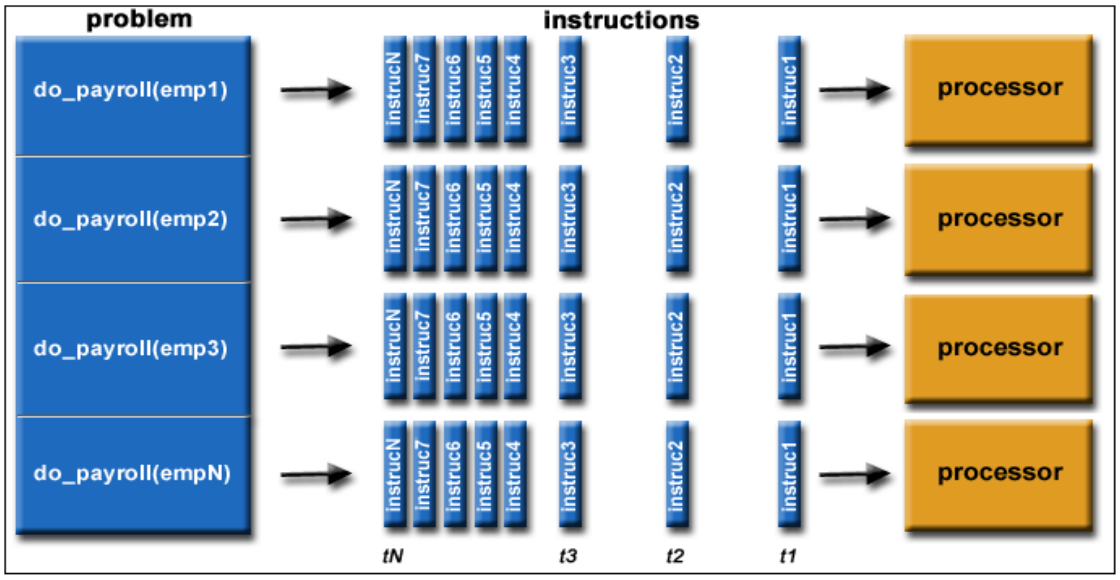


Fig.6: Computación paralela (Blaise Barney, 2015).

- El problema computacional debe ser capaz de:
    - Romperse en pedazos discretos de trabajo que se pueden resolver de forma simultánea.
    - Ejecutar múltiples instrucciones del programa en cualquier momento del tiempo.
    - Hay que resolver en menos tiempo, con múltiples recursos informáticos que con un único recurso de cómputo.
- Los recursos informáticos suelen ser:
    - Un solo equipo con múltiples procesadores / núcleos.
    - Un número arbitrario de ordenadores conectados por una red

(Joubert, 2002)

RaspberryPI2, por su mayor capacidad en memoria que la versión anterior, puede brindar las características deseables para poder programar en cada placa lo necesario para implementar lo referente a la computación paralela, esto sería de mucha ayuda, ya que se lograría ejecutar varias tareas de forma paralela, sin que la devolución de respuestas quede comprometida a un tiempo muy largo de espera. Esta implementación debería probarse, por lo que es necesario consumir del dispositivo muchos recursos, mismos que son necesarios para lograr procesar cada una de las tareas, lo que conlleva a una utilización mayor del procesador.

1.7 Sistemas Distribuidos.

Desde la implementación de la tecnología en las empresas a nivel mundial, el esquema de trabajo ha cambiado rotundamente, la digitalización de la información ha ayudado a las empresas a realizar trabajos mucho más rápidos, pero este crecimiento ha creado un problema para las empresas que aún mantienen el esquema de procesamiento local de su información; en la actualidad la cantidad de datos procesados en una empresa ha sufrido

un incremento de un 90 %, lo que genera más tiempo de espera para completar una operación. (Muñoz Escoí, 2013)

La mayoría de pymes en el país, mantiene su información en sus equipos locales, creando un problema de disponibilidad y procesamiento de información y en caso de falla de los equipos, existe un riesgo de pérdida permanente de información valiosa. La solución a este problema, son los sistemas distribuidos que crea un esquema, en donde la información está en diferentes lugares, pero el usuario final, ve al sistema como uno solo, sin saber la arquitectura que se está ejecutando por detrás de su estación de trabajo.

Entonces, un sistema distribuido no es más que varios ordenadores independientes conectados a una red LAN o WAN, que ofrecen a los usuarios la imagen de un sistema único y coherente, donde la información está disponible en tiempo real y puede ser procesada en un tiempo menor, ya que las tareas se dividen entre los equipos que están con menos carga de trabajo, y distribuyendo la información entre todos los computadores o dispositivos que conforman el sistema; ayudando a resguardar la información en caso de falla, por lo que la información ya no mantiene un esquema centralizado.

Un ejemplo de este tipo de sistemas, se da cuando se realiza búsquedas en internet empleando motores de búsqueda tales como Google, Bing, Yahoo, entre otros. Si este tipo de gestores de búsqueda emplearían un solo equipo para proporcionar el servicio, con la cantidad de solicitudes de búsqueda, esta tomaría un tiempo largo que se daría en horas o días. Sin embargo, al emplear una arquitectura distribuida, un usuario puede realizar una búsqueda en cuestión de segundos, ya que la arquitectura distribuida asigna varios equipos que proporcionen sus servicios y obtengan resultados en un tiempo reducido, presentándose ante el usuario como un sistema único. (Muñoz Escoí, 2013)

Como resultado de esta arquitectura, en todo sistema distribuido, se debe tener presente las siguientes características:

- Ocultación.- Los sistemas distribuidos son vistos como un sistema único ante el usuario final, ya que los usuarios no deben saber la arquitectura que esta por detrás, solo se presenta la información requerida.

- Acceso Homogéneo.- El acceso a la información es independiente del lugar desde donde se accede, para el usuario solo se presenta una misma interfaz y

resultados, no existe diferencia entre equipo o segmento que este procesando la información.

- Escalabilidad.- Los sistemas distribuidos no se pueden limitar a un número de ordenadores, el sistema puede seguir agregando más recursos para proporcionar un buen servicio y seguir creciendo de acuerdo a las necesidades.

- Disponibilidad.- La información debe estar disponible en todo momento, si uno de los módulos del sistema falla, la información debe estar replicada en otro modulo para que esté disponible para el usuario.

Los sistemas operativos actuales están centrados en la administración de los recursos locales de los equipos, y no para administrar recursos de otros equipos que estén comunicados entre sí. (Muñoz Escoí, 2013)

Para la implantación de los sistemas distribuidos, es necesario implementar una capa adicional, para poder controlar los recursos de los otros módulos del sistema; esta capa adicional es conocida como middleware, este proporciona la comunicación entre los ordenadores, sistemas operativos y las aplicaciones distribuidas, garantizando la disponibilidad de la información y proporcionando un único sistema al usuario.

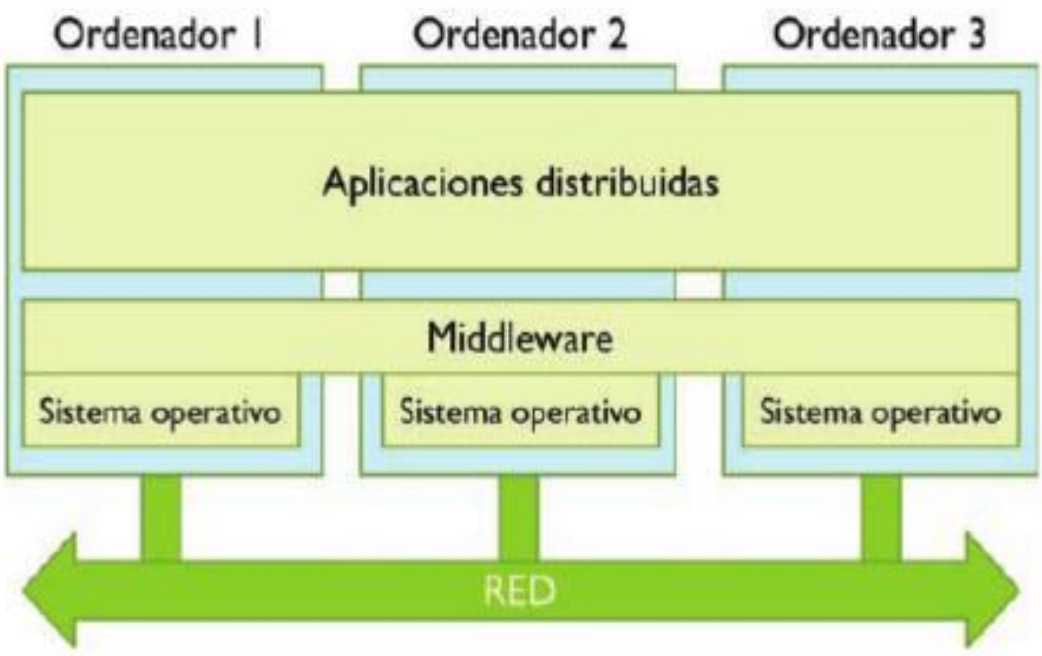


Fig.7: Sistemas Distribuidos (Muñoz Escoí, 2013).

Para lograr la implementación de los sistemas distribuidos sobre el dispositivo RaspberryPI2, es necesario implementar una capa adicional, que controle los recursos y módulos del sistema, esto permitirá tener una mejor funcionalidad del clúster, además de un mejor manejo de los recursos y una mejor predisposición de las tareas. Cada una de estas mejoras se verá reflejada en el desempeño del dispositivo RaspberryPI, ya que cada placa asignada a esta implementación, funcionaría como un intermediario entre el cliente y la petición, mejorando el tiempo de respuesta, y en la optimización de los recursos, ya que solamente se accedería a la información alojada en un servidor, sin importar su localización física.

CAPITULO 2: Tecnologías Aplicadas

2.1 Introducción

Con un enfoque al sistema que se pretende realizar y teniendo en cuenta que el tema es sobre la investigación de un dispositivo que en nuestro medio no está implementado. *RaspberryPi2*, es un dispositivo tecnológico con una capacidad de desarrollo enorme en distintas áreas de la tecnología moderna (depende mucho del medio en donde se lo quiera implementar). Por lo concerniente a la investigación a realizar será en el área de redes y telecomunicaciones, en un sistema Cliente-Servidor, que cuente con las mismas o mejores características que los servidores tradicionales, dando a las pequeñas y medianas empresas la posibilidad de contar con este sistema que les ayude a su desarrollo.

Uno de los problemas más grandes y más comunes en la implementación de nuevos dispositivos en el mercado empresarial, es que el dispositivo a implementar puede sufrir daños, ya que se va a experimentar con él, instalando diferentes aplicaciones tecnológicas que ayuden de una mejor manera a conseguir el objetivo “en la caso de investigación propuesta será de redes y de servidores”.

El desarrollo de aplicaciones Cliente-Servidor en otros países con la utilización *RaspberryPi2* es muy grande, ya que cuenta con la tecnología necesaria para funcionar

como un dispositivo para conectar maquinas en red, hasta tener la posibilidad de crear un servidor de altas características y prestaciones por su tecnología y capacidad. Es por ello que es necesario aprovechar de la mejor manera posible su tecnología y lograr crear un sistema cliente-servidor capaz de ser implementado en cualquier empresa.

2.2 Dispositivo Raspberrypi2

Raspberrypi2 es un ordenador de placa reducida, que fue creada para la enseñanza de computación, con el tiempo y luego de varias pruebas de este dispositivo, ha ido innovando y creando nuevas alternativas de desarrollo, así como de temas de investigación en las ramas de electrónica y redes de telecomunicaciones etc. (Pi, Raspberry Pi, historia Informática, 2013)

RaspberryPi en sus inicios carecía de la potencia necesaria para crear nuevas aplicaciones de desarrollo en el campo de la educación; con el paso de los años ha ido mejorando hasta la creación del nuevo Raspberrypi2, el lanzamiento de este dispositivo que tiene la virtud de solucionar las deficiencias que éste a sus inicios presentaba, ha tenido una de las mejores acogidas en el mercado educativo con fines de enseñanza, ayudando a los investigadores a la creación de nuevas alternativas de desarrollo . (Pi, Raspberry Pi, historia Informática, 2013)

El proyecto RaspberryPi fue creado por un grupo de estudiantes universitarios de Cambridge en el año 2006, pero no fue lanzado al mercado inmediatamente. En sus inicios fue creado con la intención de enseñar a niños a que desarrollaran las destrezas y despertar sus habilidades en lo que se refiere a electrónica y computación, los esquemas de diseño y características fueron pensados en este tipo de usuarios, por lo que carecía de potencia y tenía muchas limitaciones de desarrollo. (Pi, Raspberry Pi, historia Informática, 2013)

Desde la creación RaspberryPi, su innovación estaba centrada en la niñez y su objetivo seguía centrado en la creación de ordenadores de bajo costo y portables, que permitan a los niños familiarizarse con ellos y que los incentive a realizar investigaciones, de esta manera motivar en ellos su interés por la ciencia. Eben Upton, el responsable de la arquitectura de RaspberryPi, se puso en contacto con varios profesores informáticos,

especialista en la educación básica y la estimulación de niños, para que ayudaran a desarrollar una arquitectura portable y que garantice la enseñanza a los niños. (Pi, Raspberry Pi, historia Informática, 2013)

La arquitectura de desarrollo de RaspberryPi cada día está más centrada y desarrollada para aplicar tecnologías de punta, utilizadas actualmente en otros dispositivos que han dado muy buenos resultados por su visión futurista, tales como las aplicaciones utilizadas para los dispositivos móviles y portátiles, que utilizan arquitecturas ARM Raspbian (derivada de Debian), Arch Linux y Risc Os, con lo que se intenta promover el aprendizaje y desarrollo mediante herramientas OpenSource como: Python, C,C++,NetBeans etc. (Pi, Raspberry Pi, historia Informática, 2013)

Los primeros diseños de RaspberryPi basados en la arquitectura ARM fueron implementadas en un prototipo del tamaño de una memoria USB, contaba con un puerto HDMI y un puerto USB. Con el paso de los años y la continua investigación de desarrollo de RaspberryPi en el año 2011 se fabricaron como prototipos 50 placas Alpha del modelo inicial, el modelo A, 25 placas modelo B, estas fueron probadas y ensambladas en un total de 100 placas. (Pi, Raspberry Pi, historia Informática, 2013)

Lanzadas como placas de prueba y para ver la acogida que éstas tenían, se pusieron en subasta 10 placas en la tienda virtual eBay. Dando como resultado un colapso en los servidores web por las continuas actualizaciones de la página por parte de personas interesadas en comprar la placa de prueba. (Pi, Raspberry Pi, historia Informática, 2013)

Debido a la prometedora acogida y a un potencial mercado que demandaban las placas, ya no sólo por niños, sino por profesionales universitarios, con proyecto de desarrollo investigativo en diversas ramas tecnológicas, se fabricó en Taiwán y China 10000 placas de lanzamiento oficial. La prioridad de la fundación RaspberryPi ahora era abaratar costos de producción y reducir al máximo los tiempos de entrega, con este ahorro la fundación podría centrarse plenamente en invertir más dinero en lo referente a investigación y desarrollo de su producto. (Pi, Raspberry Pi, historia Informática, 2013)

La primera venta oficial fue el 29 de febrero del 2012, del modelo B de RaspberryPi. Las tiendas encargadas de la venta del dispositivo: Premier Farnell, Rs Component, rápidamente tuvieron demasiada carga en sus servidores después del lanzamiento oficial. En el primer semestre las ventas llegaron a unas 500000 unidades, siendo un

éxito y teniendo una acogida mucho mayor de la esperada. (Pi, Raspberry Pi, historia Informática, 2013)

Con la continua acogida del dispositivo, la fundación RaspberryPi decidió llevar las placas a una fábrica de la empresa Sony en reino Unido, en ellas, se producirían cerca de 30000 unidades cada mes, ayudando así al sector económico, al dar paso a la creación de nuevas plazas de trabajo. (Pi, Raspberry Pi, historia Informática, 2013)

En el primer trimestre del año 2013, se hizo el lanzamiento oficial del modelo A de RaspberryPi, que contaba entre sus características principales con tan solo 256 Mb de RAM y sin un Puerto Ethernet a un precio mucho más asequible que el modelo B. En ese año se (Pi, Raspberry Pi, historia Informática, 2013)contaba con dos modelos diferentes de RaspberryPi; Modelo A, B. El modelo A se diferenciaba del modelo B, en que carecía de un controlador Ethernet, un solo puerto USB, y 256MB de RAM, frente a los 512MB, 2 puertos USB, controlador Ethernet 10/100 con los que contaba el modelo B. (Pi, Raspberry Pi, historia Informática, 2013).

La fundación RaspberryPi, en el afán de crecer con su dispositivo frente a una PC de escritorio o portátiles, a más de su reducción en porte, costo, se ha visto en la necesidad de hacerlo un dispositivo que permita la instalación de cualquier sistema operativo. La mayoría de las computadoras portátiles de hoy en día funcionan con un sistema operativo como: Windows, Apple OS X, ambas plataformas son de código cerrado, es decir que no se puede hacer modificaciones, estas restricciones no permiten la implementación de nuevo código que ayude al desarrollo e implementación de nuevos proyectos investigativos. (Pi, Raspberry Pi, historia Informática, 2013)

El código cerrado que brindan estos sistemas operativos, no dan la posibilidad de ver cómo está hecho el software implementado sino que restringe al usuario a utilizarlo de acuerdo a lo ya previamente establecido sin dar la posibilidad de modificarlo. Por lo contrario RaspberryPi está diseñado para la utilización de código abierto, en este dispositivo se puede implementar un sistema operativo GNU/Linux, esto garantiza la descarga del código fuente del sistema operativo y adaptarlo de acuerdo a las necesidades del usuario, en este sistema operativo nada está oculto, todo está disponible para el público, ayudando así al desarrollo de la investigación y al espíritu desarrollador que los usuarios de este dispositivo RaspberryPi tienen, lo que garantiza mejorar su desempeño en su implementación. (Pi, Raspberry Pi, historia Informática, 2013)

Cabe recalcar que en el chip BCM2835 que RaspberryPi tiene como característica, vienen incorporadas varias versiones de Linux, que son unas versiones de código abierto conocidas como distribuciones donde incluye Debian, Fedora Remix y Arch Linux, estas versiones tienen como finalidad ayudar o suplir las distintas necesidades de los usuarios y además ayudar a mejorar sus aplicaciones, ya que todas son compatibles entre sí, es decir; una aplicación de software escrito en el sistema Arch Linux, funciona perfectamente en Fedora Remix y viceversa. (Pi, Raspberry Pi, historia Informática, 2013)

La arquitectura que maneja este dispositivo, es la misma que está siendo utilizada por los dispositivos móviles con sistema operativo android, que es la arquitectura ARM y la x86, creando una diferencia entre los demás sistemas operativos como: Windows, OS X y Linux. La principal diferencia es que cuando el software está escrito para Windows, no funciona para Linux, pero afortunadamente existen herramientas que ayudan hacer compatibles el software entre estos distintos sistemas operativos, pero lo mejor es que casi todas estas herramientas son de libre uso y de código abierto, como lo es el propio sistema Operativo GNU/Linux. (Pi, Raspberry Pi, historia Informática, 2013)

En la actualidad todos los sistemas operativos son compatibles con el RaspberryPi, ya sean estos los sistemas operativos completos o versiones modificadas, siendo esta una de las principales características de este dispositivo, en el entorno de escritorio utiliza un código abierto open source licenciado por GPL para Linux y otras plataformas de desarrollo POSIX compatibles. (Pi, Raspberry Pi, historia Informática, 2013)

En sus inicios, el RaspberryPi modelo B en su interior contaba con un procesador multimedia Broadcom BCM2835 system-on-chip (SoC), por lo que todos los componentes de la placa como: CPU y la GPU junto con el audio y Hardware, se encuentran integrados en un único componente debajo del chip, en la parte central de la placa. Además, cuenta con la arquitectura ARM que es una arquitectura de instrucciones distintas y open source. Todos y cada uno de estos componentes se han ido mejorando a través de la investigación, y del desarrollo de nuevas tecnologías que junto con la evolución de RaspberryPi, se ha podido desarrollar un dispositivo con mejores prestaciones y característica que los antes mencionados; el *RaspberryPi2* y la nueva arquitectura ARMv7 han conseguido proporcionar estas mejores características (Pi, Raspberry Pi, historia Informática, 2013).

Cuadro comparativo de los diferentes modelos RaspberryPi. (Pi, Raspberry Pi, historia Informática, 2013)

	RPI A	RPI A+	RPI B	RPI B+	RPI B 2
SoC	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836
CPU	ARM11 ARMv6 700MHz	ARM11 ARMv6 700 MHz.	ARM11 ARMv6 700 MHz.	ARM11 ARMv6 700 MHz.	ARM11 ARMv7 ARM Cortex-A7 4nucles a 900 MHz
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
RAM	256 MB LPDDR SDRAM 400 MHz	256 MB LPDDR SDRAM 400 MHz	512 MB LPDDR SDRAM 400 MHz	512 MB LPDDR SDRAM 400 MHz	1 GB LPDDR2 SDRAM 450 MHZ
USB 2.0	1	1	2	4	4
Salida de Video	HDMI 1.4 a 1920x1200 pixeles	HDMI 1.4 a 1920x1200 pixeles	HDMI 1.4 a 1920x1200 pixeles	HDMI 1.4 a 1920x1200 pixeles	HDMI 1.4 a 1920x1200 pixeles
Storage	SD/MMC	microSD	SD/MMC	microSD	microSD
Ethernet	No	No	Si, 10/100 Mbps	Si, 10/100 Mbps	Si, 10/100 Mbps
Tamaño	85.60x56.5 mm	65x56.5 mm	85.60x56.5 mm	85.60x56.5 mm	85.60x56.5 mm
Peso	45 g.	23 g.	45 g.	45 g.	45 g.
Precio	25 dólares	20 dólares	35 dólares	35 dólares	35 dólares

Fig.8: Comparación de los modelos RaspberryPIs (Pi, Raspberry Pi, historia Informática, 2013) .

2.3 Procesadores ARMv7

Una de las características fundamentales de los procesadores ARM es que todas sus partes funcionan como una sola y que los programas que se compilen lo hagan de una forma correcta. El rendimiento es la principal característica en los procesadores ARM, que utilizan una arquitectura RISC (Reduced Instruction Set Computer), esta era una de las principales características que distinguía a este procesador de cualquier otro desde sus inicios, cuando se hizo su lanzamiento de sus primeros prototipos en el año de 1985. (Dang, 2014)

Esta arquitectura hoy en día esta sobre otras arquitecturas existentes y a la vanguardia por sus bondades y características, sobre todo por estar en una continua investigación a la par del desarrollo de los nuevos dispositivos móviles, especialmente los de telefonía como la marca Samsung, que utiliza actualmente esta arquitectura en casi todos sus dispositivos. (Dang, 2014)

Las bondades de estos procesadores han llevado a que todos los programas puedan ser compilados en ordenadores con características muy básicas, es por ello de la utilización de RaspberryPi con sus distribuciones estándares de GNU/Linux, su operabilidad permite que estos sistemas funcionen con cualquier dispositivo moderno. La empresa encargada de desarrollar esta tecnología es una empresa inglesa también llamada ARM, que ofrecen esta tecnología a Grupos como Freescale, TI, Samsung, entre otras. Y estos a su vez los modifican de acuerdo a sus necesidades para así sacarles su máximo rendimiento y con un menor consumo de energía. (Dang, 2014)

En ordenadores, tanto Intel como AMD, utilizan la misma arquitectura desarrollada por Intel (llamada x86), en los dispositivos móviles casi el 98% de ellos emplean actualmente la arquitectura ARM. Esta arquitectura la define como una compañía que no hace chips (ARM). Los chips son elaborados por compañías que tienen su propia arquitectura, notoriamente esta tecnología tiene cada día más avances, y como en todo articulo o accesorio tecnológico, las arquitecturas más antiguas utilizan los chips de implementación más baratos. (Dang, 2014)

La empresa Acorn Computers, desarrolló la arquitectura RISC de 32 bits en el año de 1980, cinco años antes de hacer el lanzamiento de los primeros prototipos ARM, la arquitectura fue de mucha ayuda y de utilidad, pese a su limitación en productos y disponibilidad, años después la compañía llamada ARM holdings se formó para licenciar la arquitectura para su uso, y así dar paso a la creación de nuevos productos, esta tecnología ya era utilizada en electrónica, en las áreas de la mecánica automotriz, reproductores de música, televisores, etc., que vieron en estos dispositivos unas características únicas y de amplio aprovechamiento en la investigación de nuevas aplicaciones. (Dang, 2014)

ARM desarrolló una serie de núcleos bien definidos y específicos (ARM7, ARM7TDMI, ARM926EJS, corteza), esto no debe llevar a confundir con las diferentes especificaciones de la arquitectura que son: ARMv1-ARMv7. Si bien actualmente los dispositivos utilizan la arquitectura ARM pero utilizan distintas versiones dependiendo del dispositivo, ya que no es lo mismo tener un Samsung Young que utiliza la arquitectura ARMv4 que un Samsung S5 que utiliza una arquitectura ARMv7. (Dang, 2014)

Los procesadores ARMv4 y ARMv5 son las primeras versiones de la arquitectura ARM, es decir son relativamente caducos para la actualidad, pero cabe recalcar que son versiones dominantes dentro del diseño con más de 10 millones de existencias de acuerdo con el centro de comercialización de ARM. Actualmente productos electrónicos que son de consumo popular, como los celulares, utilizan versiones más recientes de la arquitectura, ejemplo., la tercera generación de Apple iPod Touch y iPhone ejecuta en un chip ARMv6, y más tarde iPhone / iPad y Windows Phone 7 dispositivos están todos en ARMv7. Mientras que las empresas como AMD e Intel sigue un patrón de diseño de fabricación ligeramente en sus procesadores ARM, es decir ARM diseña la arquitectura y licencias para que otras empresas lo integren de acuerdo a sus necesidades en los dispositivos desarrollados. Empresa como Apple, NVIDIA, Qualcomm y Texas Instruments comercializan sus propios procesadores (A, Tegra, Snapdragon, y OMAP, respectivamente), pero su arquitectura central tiene licencia de ARM. Todos estos implementan el modelo de instrucción previamente establecido y todos se definen mediante el manual de referencias de ARM. Cuando ellos quieren agregar extensiones adicionales al procesador, lo pueden hacer por ejemplo, la extensión Jazelle permite que bytecode de Java se ejecute de forma nativa en el

procesador. La extensión del pulgar añade instrucciones que pueden ser 16 o 32 bits de ancho, lo que permite mayor densidad de código (instrucciones ARM nativo son siempre 32 bits de ancho). (Dang, 2014)

Cada uno de estas extensiones son de depuración, es decir cada empresa utiliza esta arquitectura y luego hace pruebas para mejorar su dispositivo, tanto en tiempo de respuesta como en consumo de energía, analizan el procesador físicamente utilizando herramientas de hardware especialmente diseñado para depuración, cada una de estas extensiones lleva una letra única que las identifica (A,B,C,D..J,K..etc). Los fabricantes son los que determinan si estas extensiones necesitan de una licencia o no, dependiendo de su utilización y de sus necesidades. Es por ello que ARMv6 y procesadores anteriores tienen letras después de ellos, ejemplo, ARM1156T2 significa ARMv6 con Thumb-2 de extensión. Todos estos acuerdos establecidos en estas arquitecturas en ARMv7 ya no se utilizan, en su lugar utiliza la aplicación, tiempo real, microcontroladores y el nombre del modelo con diferentes características, ejemplo, ARMv7 Cortex-A serie son los procesadores con el perfil de aplicación; y Cortex-M son para microcontroladores y único apoyo ejecución modo Pulgar. (Dang, 2014)

2.3.1Características Básicas ARM

ARM es una arquitectura RISC, existen varias diferencias entre ARM y la arquitectura CISC (x86 / x64). Las nuevas versiones de los procesadores Intel cuentan con alguna arquitectura RISC y no son puramente CISC, es decir que el conjunto de instrucciones ARM es muy pequeño en comparación con x86, pero ofrece una mayor variedad de registros de propósito general, otra característica es que la longitud de instrucciones fija el ancho (16 o 32 bits), esto depende mucho del estado en el que se encuentre. Además que ARM utiliza el modelo carga-almacenamiento para acceso a memoria. Es decir que los datos deben ser movidos a memoria antes de ser operados, y solo las instrucciones carga/almacenaje pueden acceder a la memoria. En ARM esto se traduce a instrucciones LDR (load a word from memory) y STR(store a word to memory). Es decir, si se desea incrementar un valor de 32 bits en cualquier dirección de memoria en particular, primero se debe cargar dicho valor en esa dirección a un registro, incrementarlo y luego almacenarlo de nuevo. A diferencia de x86, que permite que la

mayoría de sus instrucciones operen directamente con los datos en la memoria, una operación tan sencilla en la arquitectura ARM requiere de 3 instrucciones (uno de carga, uno de incremento, y otra de almacenaje) esto puede conllevar a que haya más código que leer en la aplicación de la ingeniería inversa, pero ya en la práctica e implementación esto no importa mucho. (Dang, 2014)

Dentro de la implementación de privilegios, en x86 cuenta con 4 privilegios que van desde el 0-3, siendo el 0 el de mayor privilegio y el 3 con el menor. ARM también cuenta con privilegios, 8 en total, estos privilegios se definen de modos diferentes:

ARM también ofrece varios niveles de privilegio para implementar el aislamiento. En x86, los privilegios se definen por cuatro anillos, con el anillo 0 tener el mayor privilegio y el anillo 3 que tiene el más bajo. En ARM, los privilegios se definen por ocho modos diferentes:

- User (USR)
- Fast interrupt request (FIQ)
- Interrupt request (IRQ)
- Supervisor (SVC)
- Monitor (MON)
- Abort (ABT)
- Undefined (UND)
- System (SYS)

Cada uno de los códigos que se ejecutan tienen determinados privilegios y algunos no cuentan con estos privilegios, es decir, si el código que se ejecuta en USR no puede modificar los registros del sistemas, eso lo puede modificar solo el código en modo SVC. USR es el método menos privilegiado, si bien las diferencias técnicas son muchas se puede decir que el USR tiene como privilegio el anillo 3 y SVC es como un anillo 0. La mayoría de los sistemas operativos que utilizan esta arquitectura implementan el modo kernel en SVC dándole acceso a todo el hardware y toda la

memoria, en cambio el modo de usuario en USR carece de acceso directo al hardware. Ambos sistemas operativos Windows y Linux hacen esto. (Dang, 2014)

Los procesadores x64 se pueden ejecutar en 32 bits o en 64 bits o ambas indistintamente. En los procesadores ARM este proceso es muy similar ya que puede funcionar en los dos estados ARM y Thumb. ARM/Thumb determina solo el conjunto de instrucciones, no el nivel de privilegios que este tiene, es decir, si se ejecuta un código en modo SVC este puede ser ARM o Thumb. Pero en el estado ARM, las instrucciones son de 32 bits de ancho siempre, mientras que en Thumb, las instrucciones pueden ser de 16 o 32 bits de ancho. Aquí el procesador ejecuta estas instrucciones dependiendo de dos condiciones:

- Cuando deriva con el BX y la instrucción BLX, si el bit menos significativo del registro es 1, entonces pasara el estado Thumb. Esto se lo realizará a pesar de que las instrucciones sean de 2 o 4 bytes, el procesador no toma en cuenta el bit menos significativo, y así no habrá problemas de alineamiento.

- Si el bit T en el registro de estado del programa actual (CPSR) se establece, entonces está en modo Thumb.

Se debe recalcar que, ARM además de tener diferentes estados en ejecución, también es compatible con la ejecución condicional, es decir que una instrucción codifica algunas condiciones aritméticas que deben cumplirse con el fin de que este se ejecute. Ejm, una instrucción puede especificar que solo se ejecutara, si el resultado de la instrucción anterior es de cero. (Dang, 2014)

Esto contrasta con x86, para la que se ejecuta casi cada instrucción anterior de una forma incondicional es decir que: (Intel tiene un par de instrucciones que apoyan directamente la ejecución condicional. Cmov y Setne). La ejecución condicional es muy útil y de mucha importancia, ya que reduce las instrucciones de salto que son costosas y con ello se reduce también el número de instrucciones a ejecutar, resultando en una mayor densidad de código. Todas las instrucciones en estado ARM soportan la ejecución condicional, pero estas por defecto se ejecutan incondicionalmente. En el estado Thumb, esta instrucción especial IT es necesaria para permitir la ejecución condicional. (Dang, 2014)

Una característica única de ARM es el *barrel shifter*. En donde ciertas instrucciones pueden contener otras instrucciones aritméticas que desplazan o rotan un registro, esto es útil ya que se puede reducir varias instrucciones a una sola, ejemplo, si se deseara multiplicar un registro dos veces y luego guardar el resultado en otro registro, normalmente esto requeriría dos instrucciones (una multiplicación seguida de un movimiento) pero con el *barrel shifter* se puede incluir la multiplicación (shift left by 1) dentro de la instrucción MOV, por lo tanto la instrucción quedaría de la siguiente forma:

```
MOV R1, R0, LSL # 1          ;R1 = R0 * 2 (Dang, 2014) (Yadav, 2009 )
```

CAPITULO 3: Desarrollo e implementación.

Antes de iniciar con el proceso de desarrollo e implementación, se ha visto la necesidad de puntualizar los temas que se van a tratar en este Capítulo 3, que ya han sido mencionadas con anterioridad, pero que es necesario tenerlas muy presentes. Los temas que formarán parte de este capítulo son:

- 3.1 Sistemas operativos que soporta RaspberryPi y características del dispositivo.
- 3.2 Clúster con RaspberryPi2
- 3.3 Configuración de la Red
- 3.4 Conectar una USB en el nodo maestro
- 3.5 Configuración del Servidor FTP
- 3.6 MySQL
- 3.7 Alta Disponibilidad

3.1 Sistemas operativos que soporta RaspberryPi y características del dispositivo.

Como se ha mencionado en el capítulo anterior, RaspberryPi es uno de los computadores más baratos que existe en el mercado, ya que los componentes que lo conforman son de muy bajo costo, tales como: procesadores ARMv7 procesador de cuatro núcleos, memoria RAM de 1 GB, salida HDMI, puerto Ethernet, 4 puertos USB, puerto para cámara, puerto para pantalla táctil, ranura microSD, salida de audio, puerto de 40 pins, para conexión serial y la entrada de alimentación de poder.

Ahora bien, el equipo requiere de un sistema operativo que lo controle: es por ello que RaspberryPi permite instalar varias opciones de sistemas operativos, que permitan su uso eficiente, así como lograr un control total de la placa, entre los que se puede enumerar:

- Ubuntu Mate
- OSMC
- Openelec
- Risc Os
- Windows 10 IOT Core
- Pinet
- Snappy Ubuntu Core

Este dispositivo en su tarjeta MicroSD, incorpora un núcleo conocido como Noobs, que permite instalar diferentes sistemas operativos para el manejo del dispositivo.

Noobs permite instalar el sistema operativo Raspbian, que viene por omisión con el dispositivo RaspberryPi. Este sistema operativo está basado en Linux, garantizando su libre manejo y modificación, de acuerdo a las necesidades del usuario. Noobs es un sistema no amigable con el usuario ya que su interfaz gráfica se asemeja a las versiones iniciales de Linux, es decir un sistema de aplicaciones en modo texto para interactuar entre los equipos y los usuarios. En la figura se muestran los logos tanto de Noobs, así como el logo del sistema operativo Raspbian.

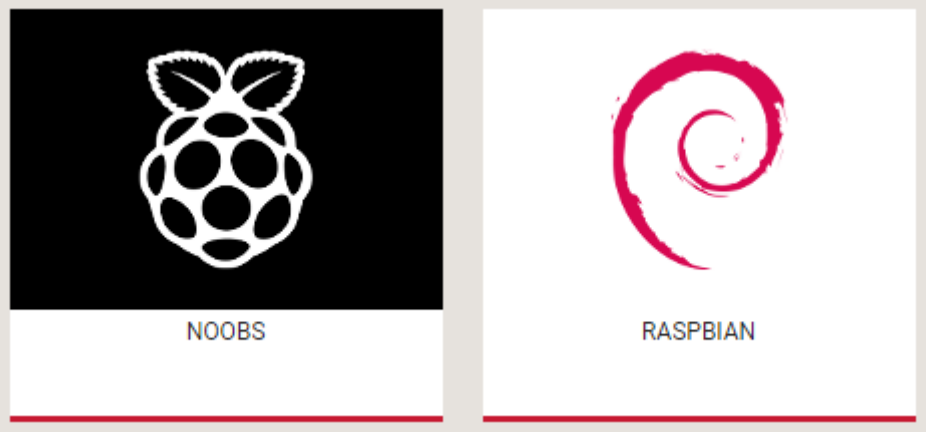


Fig.9: Logos Noobs y Raspbian.

Los sistemas operativos anteriores vienen configurados de fábrica. También existen otros sistemas operativos que pueden ser instalados en el dispositivo, dependiendo de las necesidades y del objetivo para el que se esté utilizando este dispositivo. A lo largo de esta investigación, se ha conocido varios sistemas operativos que podrían ser implementados en la placa, con la capacidad de soportarlo y funcionar correctamente. A continuación, se mostrarán algunas de los logos de los sistemas operativos:

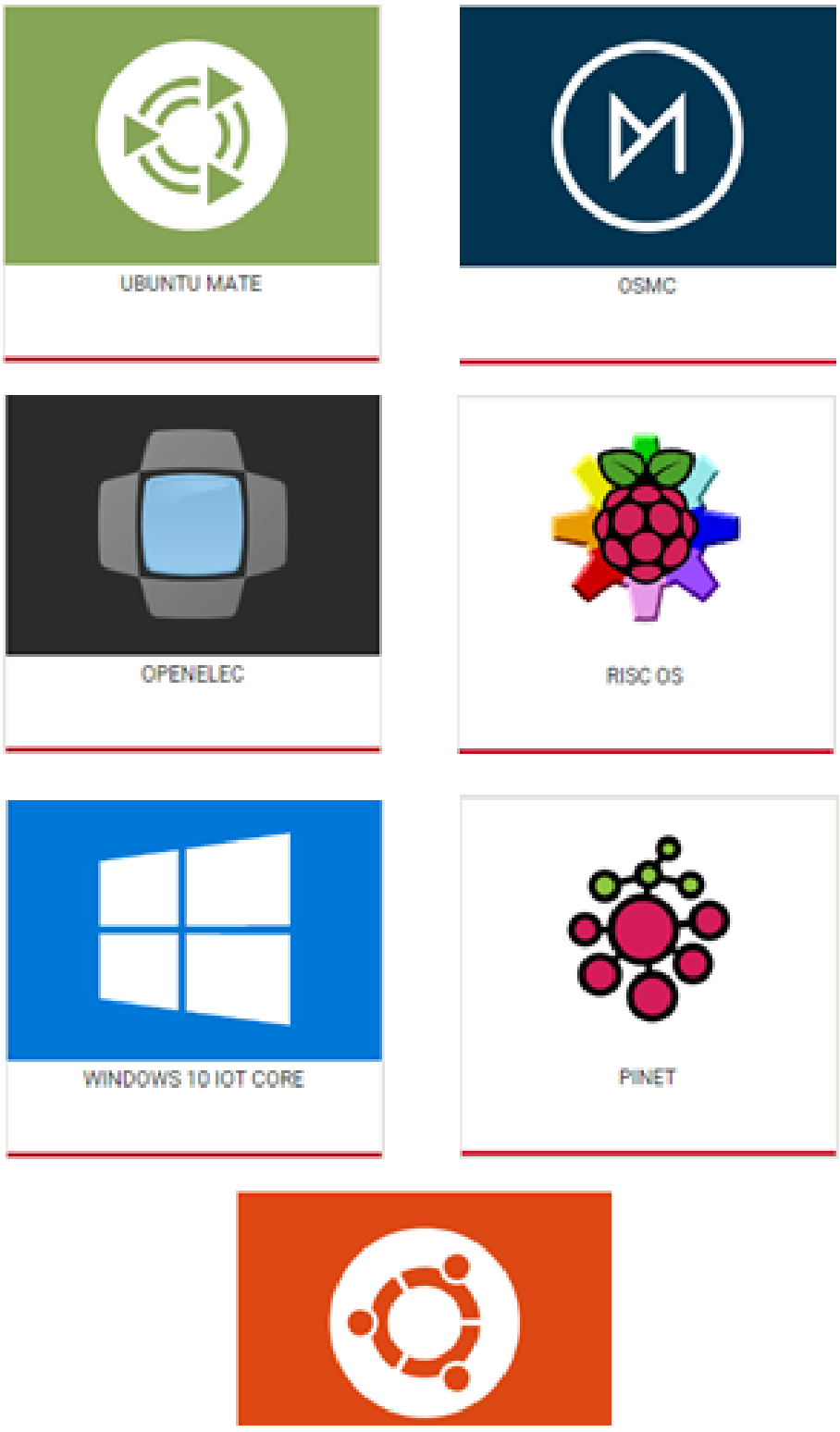


Fig.10: Logos de los Sistemas Operativos compatible con RaspberryPi.

Luego de analizar la placa y con el conocimiento de las funcionalidades que se han estudiado en el Capítulo I, acerca del RaspberryPi, se ha desarrollado la implementación del clúster tanto en hardware como en software, que permita manipular de una mejor manera cada uno de sus componentes, para así lograr un mejor desempeño y obtener los resultados esperados.

3.1.1 Instalación del Sistema Operativo

Lo primero a realizar en esta investigación, será la instalación del sistema operativo en las 3 placas de RaspberryPi. Para ello, se utilizará Raspbian basado en Linux, que ayudará a su manejo y configuración para gestionarlas de acuerdo a las necesidades. Cabe recalcar que si en el dispositivo no viene incorporada la tarjeta de memoria, se deberá incorporar una para proceder a instalar y configurar el sistema operativo Raspbian; previamente es necesario formatear la tarjeta SD.

Para que el sistema funcione correctamente, primero se debe descargar el programa SDFormatterv4 y luego ejecutarlo como administrador, como se muestra en la imagen:

Fig.11: Instalación del programa SDFormatterv4.

A continuación, se muestra la carpeta y dirección en la que se encontrará el programa que se ejecuta.

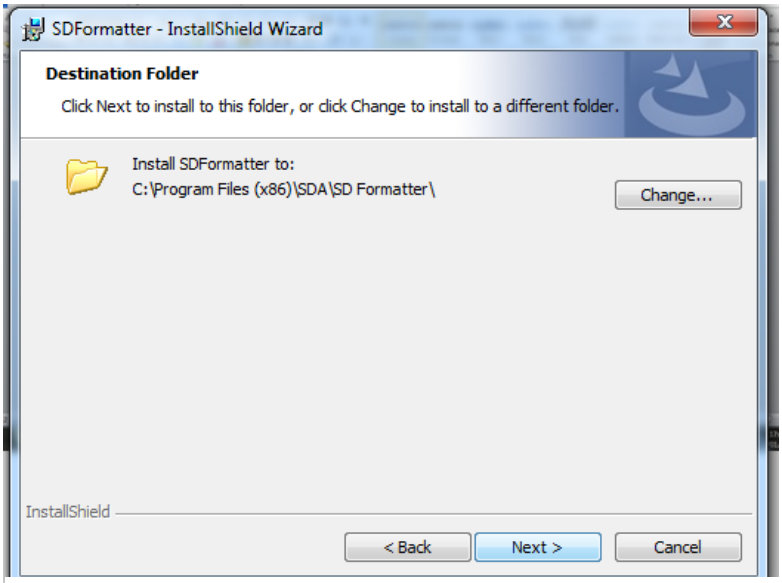


Fig.12: Instalación del programa SDFormatterv4, carpeta de destino.

Una vez copiado el instalador en la carpeta de destino indicada, se da doble clic sobre el icono de instalación y SDFormatter le dará la bienvenida.

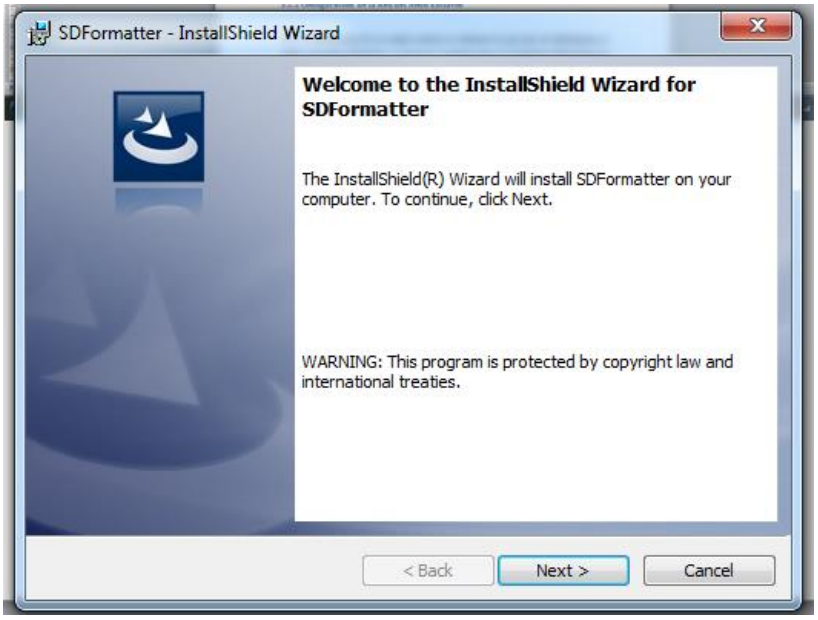


Fig.13: Instalación del programa SDFormatterv4, pantalla de bienvenida a SDFormatter.

Presione Next y expondrá una nueva imagen.

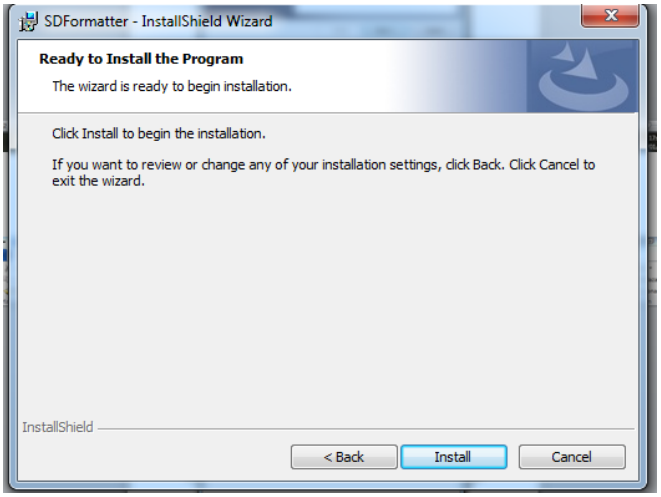


Fig.14: Instalación del programa SDFormatterv4, proceso de instalación, o cancelación del programa.

Esta pantalla es donde se instala el programa o se revierte los cambios hechos en caso de que se quiera cancelar. Para este caso de estudio se dará clic en Install.

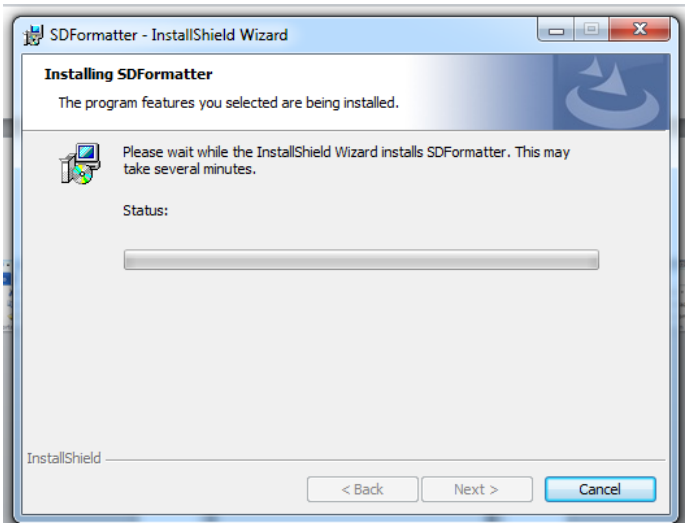


Fig.15: Instalación del programa SDFormatterv4, proceso de instalación.

Si en el proceso de instalación no existe ningún problema al momento de la instalación, se espera unos minutos hasta que la barra de progreso llegue al 100% y abrirá una pantalla como la siguiente.

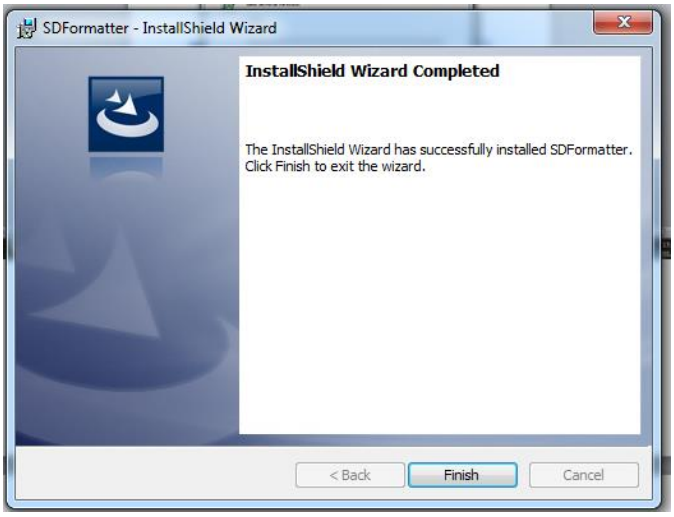


Fig.16: Instalación del programa SDFormatterv4, finalización de instalación.

Esta imagen indica la finalización del proceso de instalación del programa, que servirá para dar formato a las memorias SD que se incorporarán a los dispositivos. Esto ayudará para posteriormente instalar el sistema operativo Raspbian.

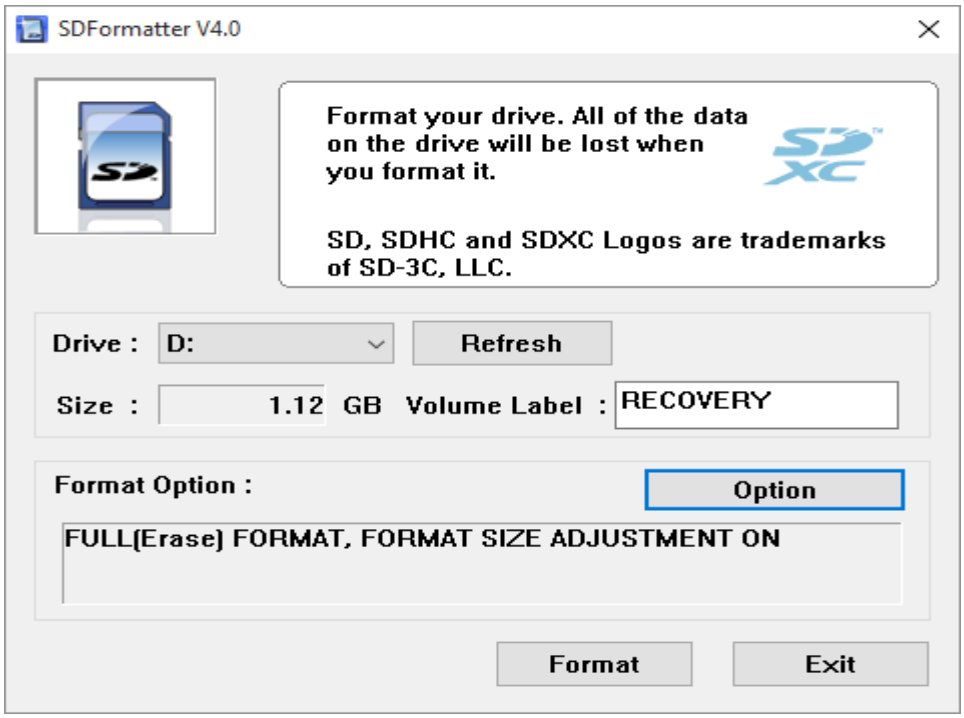


Fig.17: Instalación del programa SDFormatterv4.

Antes de abrir el programa se deberá insertar la tarjeta SD en el computador, luego se procederá a ejecutar el programa, el cual automáticamente muestra la unidad en la que está la tarjeta a la cual se le va a dar formato. Posteriormente, se da clic sobre el botón de opciones en donde revelará una pantalla igual a la siguiente.

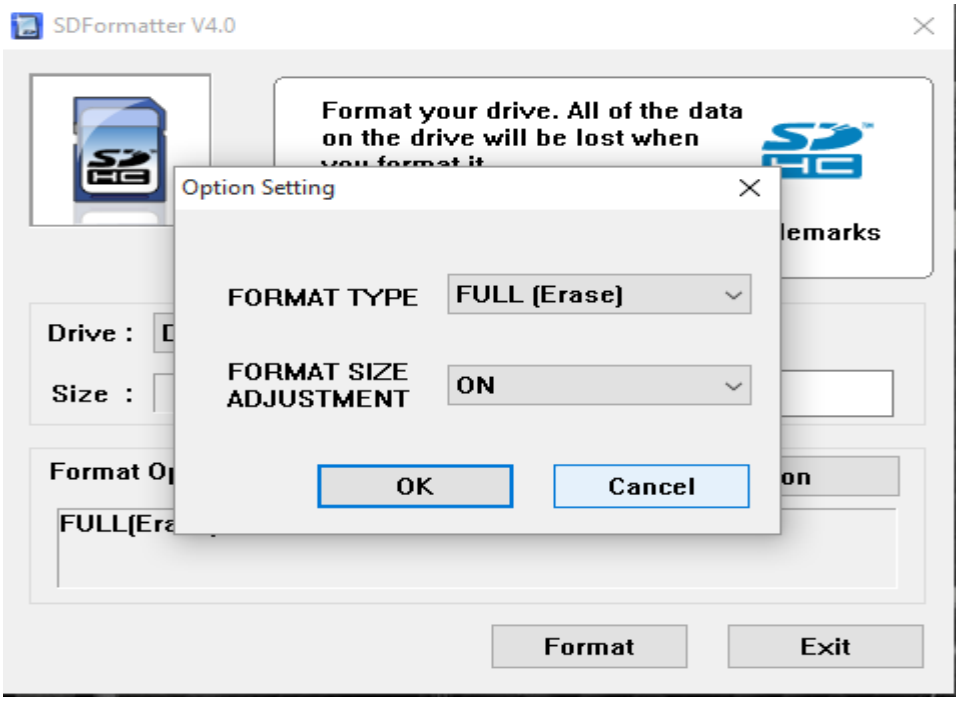


Fig.18: Como utilizar y configurar SDFormatterv4, configuración de opciones de formato.

En esta pantalla, en FORMAT TYPE, se selecciona la opción FULL(Erase) y la opción de FORMAT SIZE ADJUSTMENT= ON y se hace clic en Ok y mostrará la siguiente imagen.

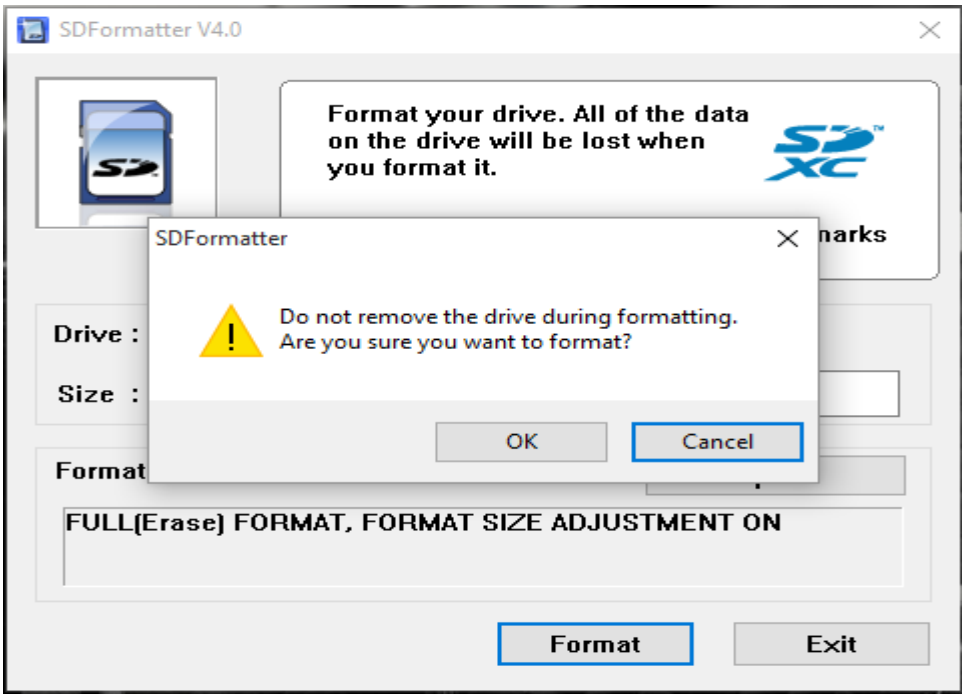


Fig.19: Como utilizar y configurar SDFormatterv4, pantalla de confirmación de dar formato a SD.

Se da clic sobre Ok y se mostrará la siguiente pantalla.

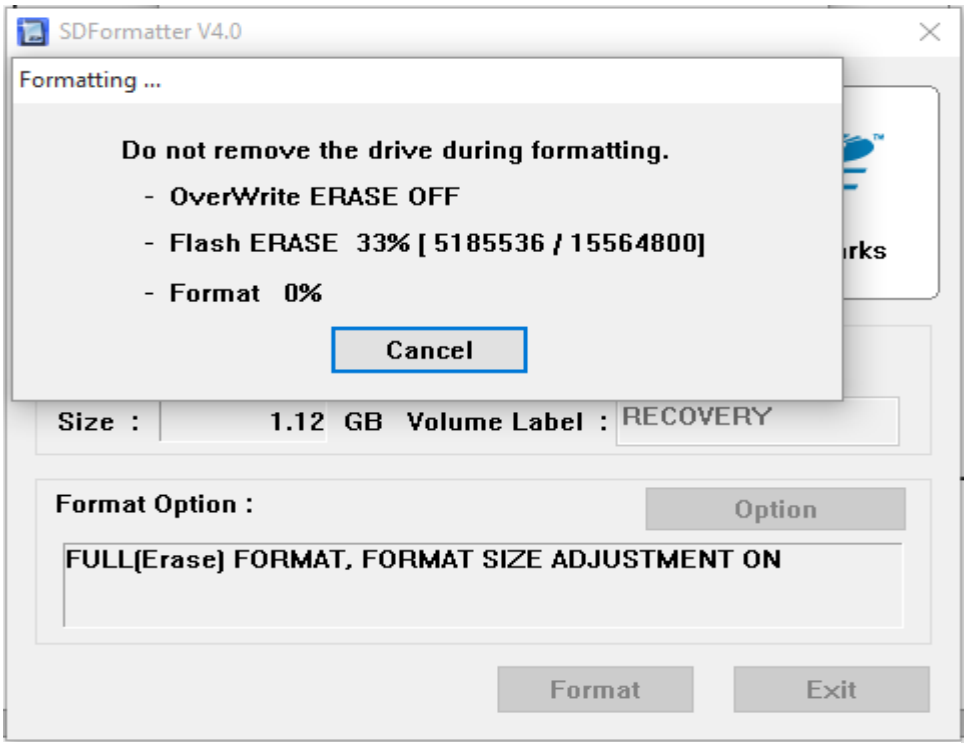


Fig.20: Como utilizar y configurar SDFormatterv4, formateando SD.

Una vez completado el formato, y si no existió ningún inconveniente durante el proceso de formateo, aparecerá la siguiente imagen mostrando que el formato se ha completado con éxito.

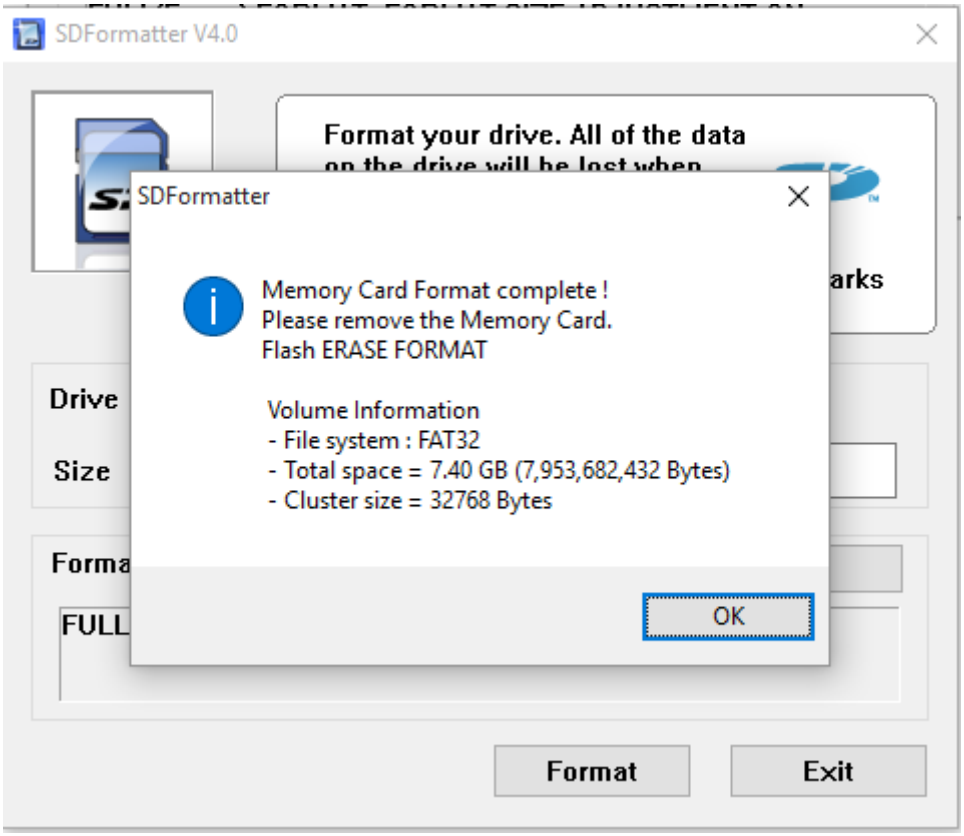


Fig.21: Como utilizar y configurar SDFormatterv4, SD formateada.

Se da clic sobre Ok, y mostrará la siguiente imagen.

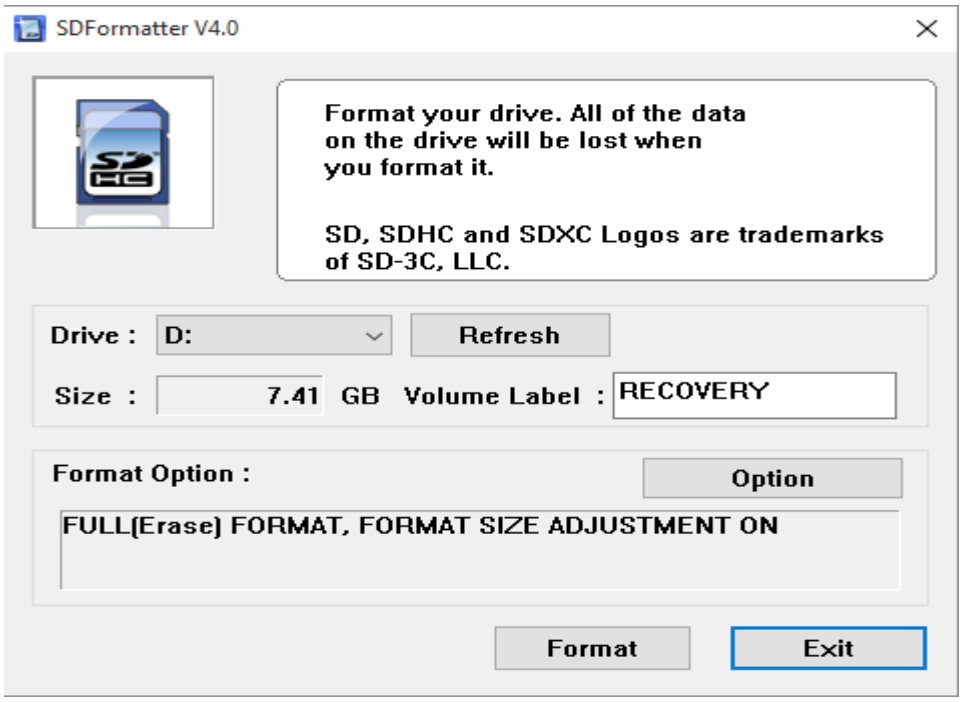


Fig.22: Como utilizar y configurar SDFormatterv4, finalización del proceso de formateo.

Una vez concluido este proceso de formateo, se iniciará el proceso de instalación del sistema operativo Raspbian sobre la tarjeta SD. A continuación se abre la unidad SD y se copia los archivos de la imagen del sistema operativo Raspbian sobre la memoria SD.

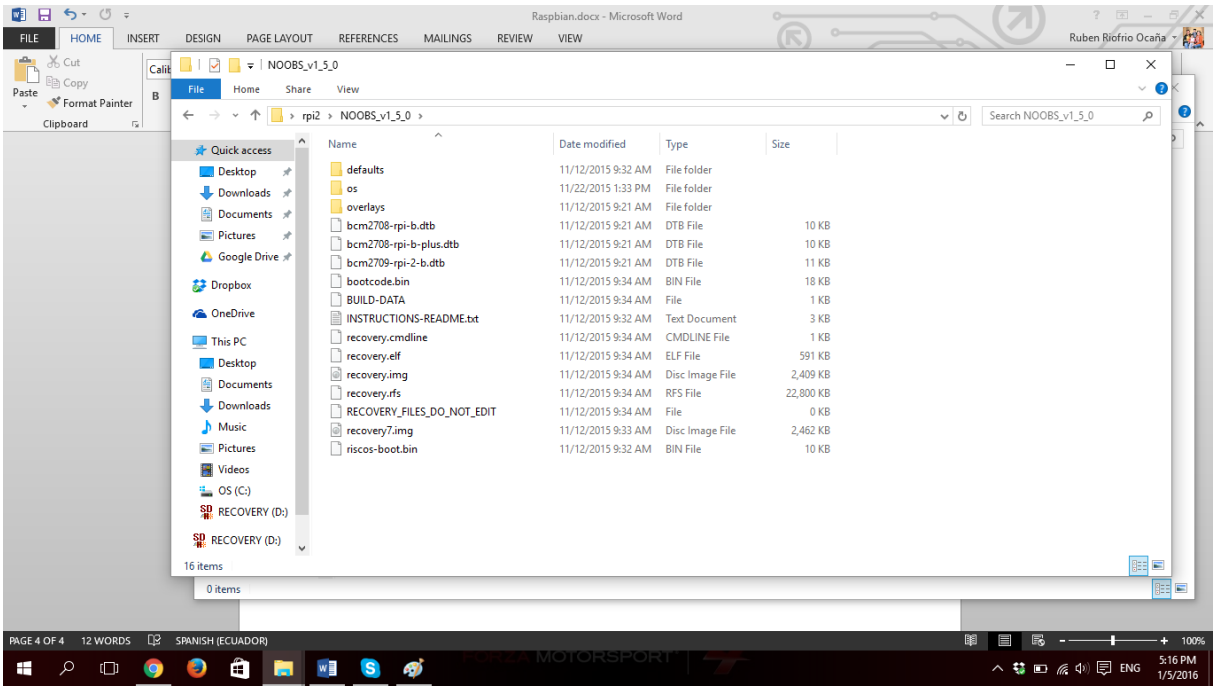


Fig.23: proceso de instalación de Rasbian en SD.

Se abre la carpeta NOOBS y se procede a copiar todos los archivos que contiene esta carpeta, luego se pega los archivos dentro de la tarjeta SD, posteriormente se elige el sistema operativo a instalar en el caso de estudio selecciona el único que aparece, que es Raspbian. En este paso se debe hacer una aclaración, que si el dispositivo RaspberryPi está conectado a internet, en la imagen se mostrarán otros sistemas operativos que se podría instalar.

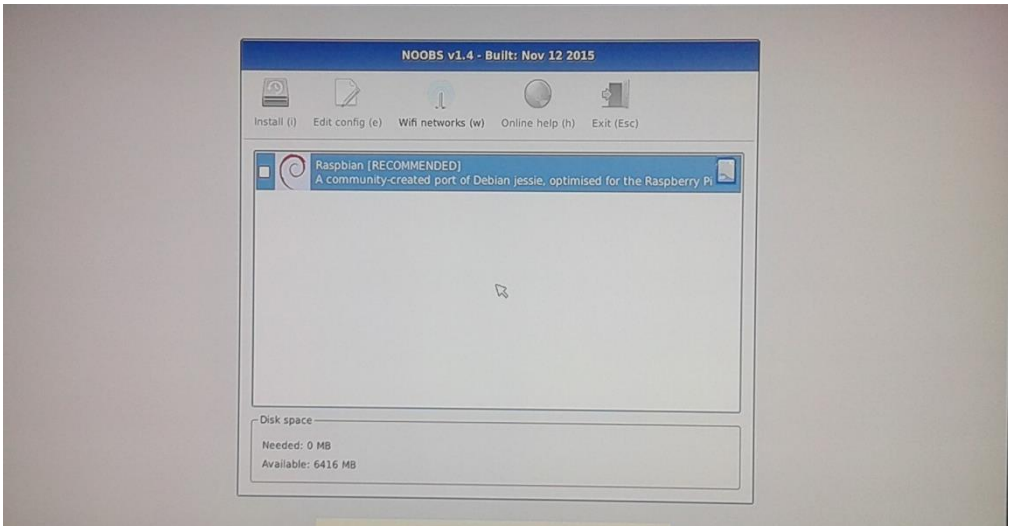


Fig.24: proceso de instalación de Rasbian en SD, selección del sistema operativo.

Una vez seleccionado el sistema operativo, se procede a elegir el tipo de idioma con el que va a ser configurado.

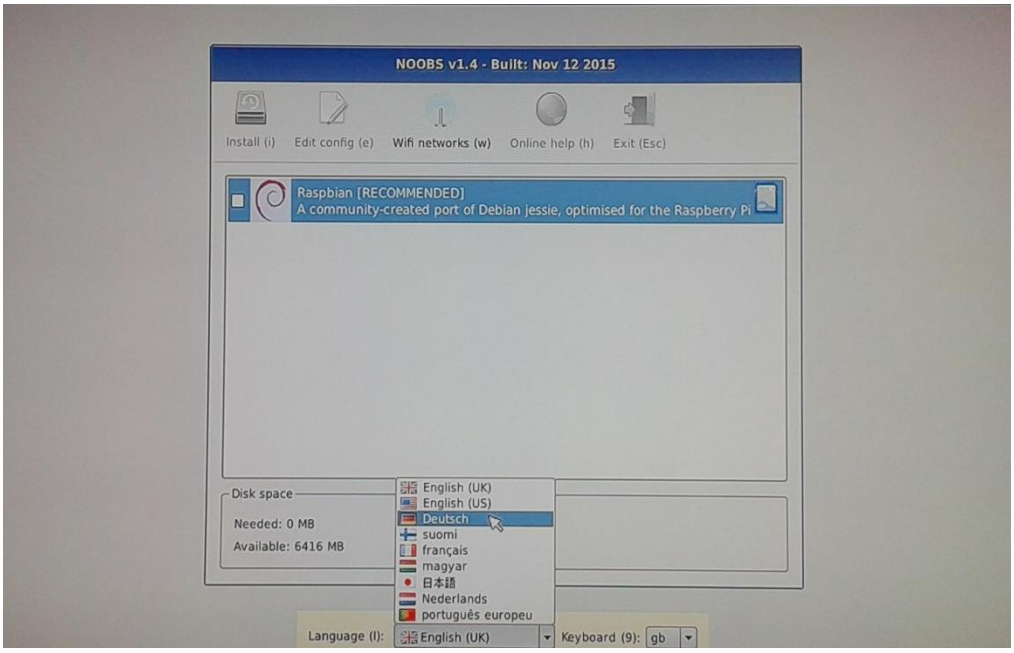


Fig.25: proceso de instalación de Rasbian en SD, selección de idioma .

Seleccionado el idioma que se va a configurar el sistema operativo, se da clic en seleccionar sobre el icono del sistema operativo Raspbian como se muestra en la figura.

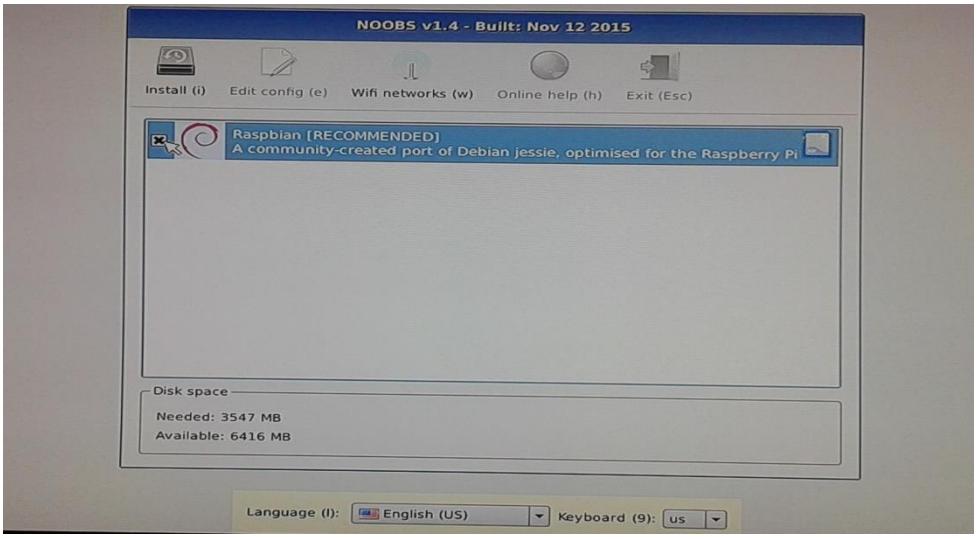


Fig.26: proceso de instalación de Rasbian en SD, opciones seleccionadas .

Una vez hecho esto se pasa al siguiente punto que es el de instalación, aquí se da clic sobre el icono install como se muestra en la figura.

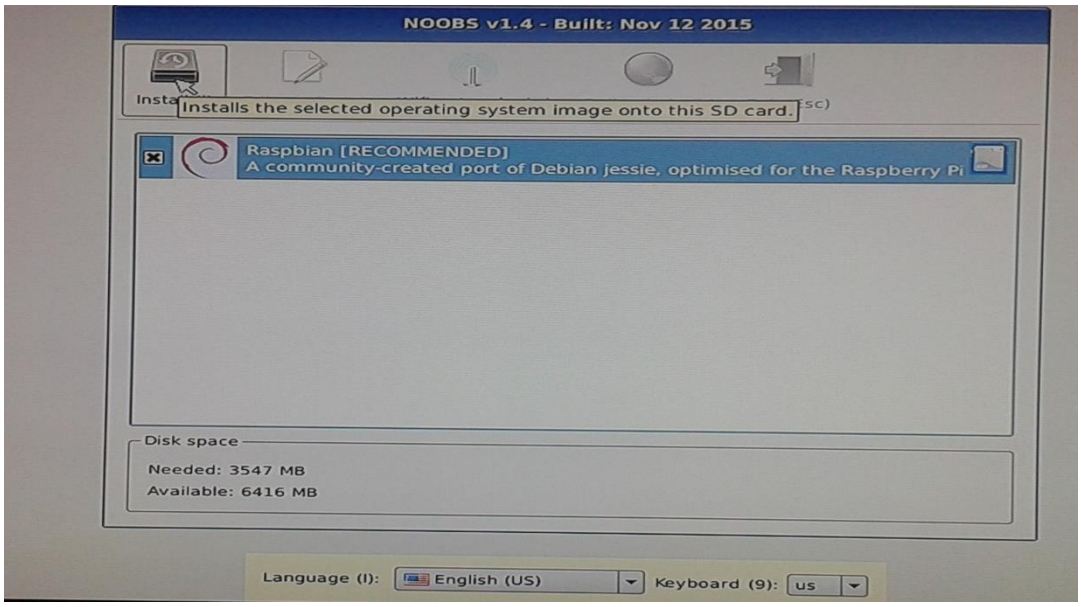


Fig.27: proceso de instalación de Rasbian en SD, instalación del sistema operativo.

Luego de esto descubrirá una imagen de advertencia de lo que se está por realizar, dando a conocer que se va a instalar un nuevo sistema operativo y que todo lo existente en la SD se borrará y se sobrescribirá sobre ella. De clic sobre Yes.

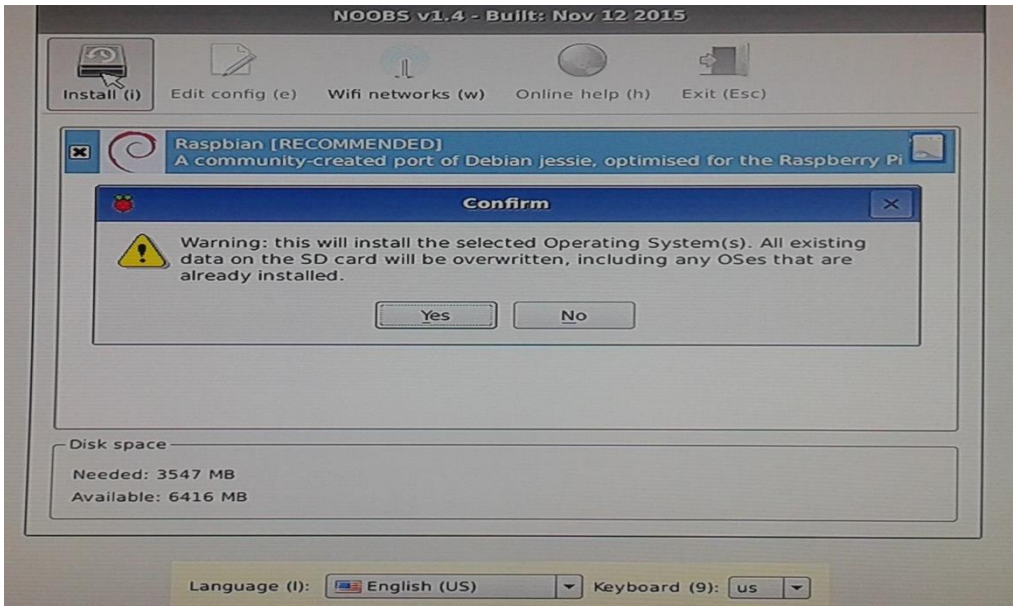


Fig.28: proceso de instalación de Rasbian en SD, confirmación de instalación de Sistema operativo en SD.

A continuación mostrará una imagen, indicando el proceso de instalación de Raspbian.

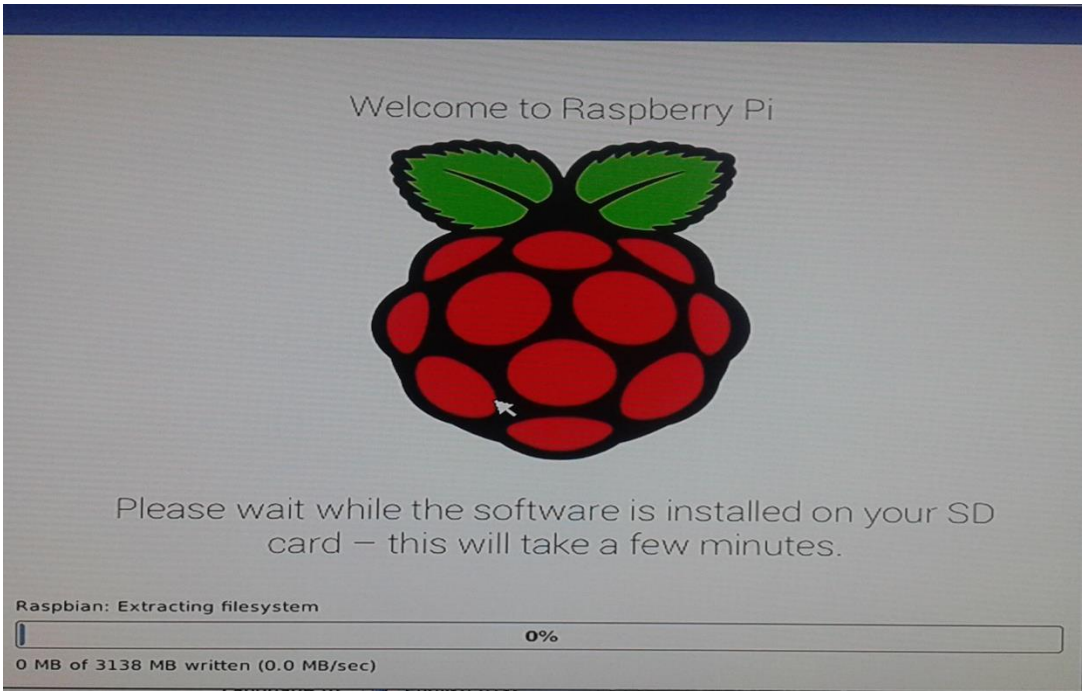


Fig.29: proceso de instalación de Rasbian en SD, instalando Raspbian.

Una vez inicializado la instalación, durante todo este proceso mostrará imágenes de las aplicaciones que se puede realizar con este dispositivo y en qué áreas se lo podría utilizar.

Al concluir con este proceso mostrará una pantalla indicando la finalización del mismo y que la instalación se realizó satisfactoriamente.

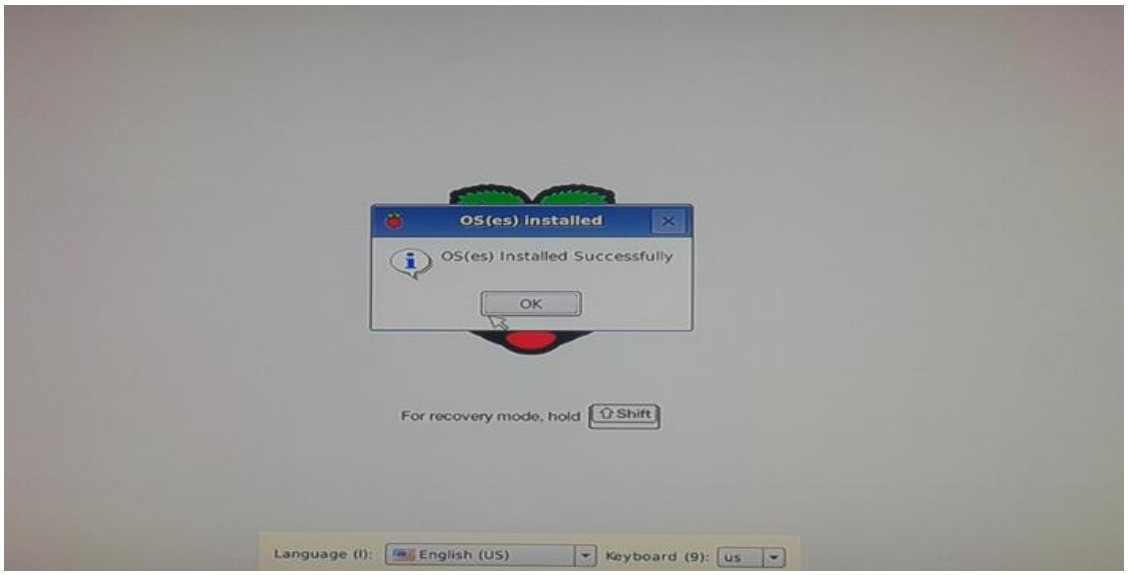


Fig.30: proceso de instalación de Rasbian en SD, finalización de instalación de Raspbian.

Se da Clic sobre ok y se mostrará los comandos de inicialización del sistema operativo Raspbian.

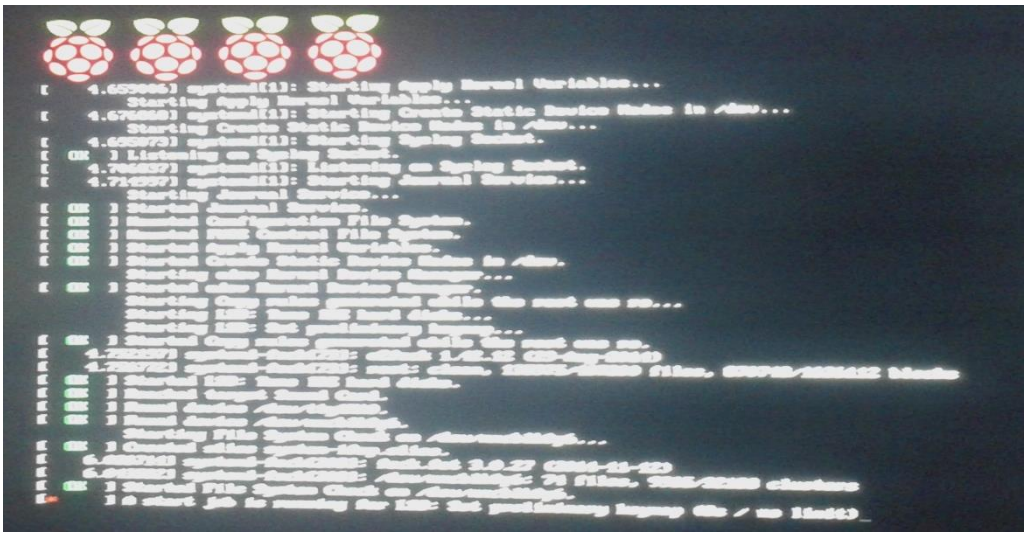


Fig.31: proceso de instalación de Rasbian en SD, configuración de escritorio de Raspbian.

Una vez terminada este proceso mostrará el escritorio del sistema operativo como se muestra en la figura.

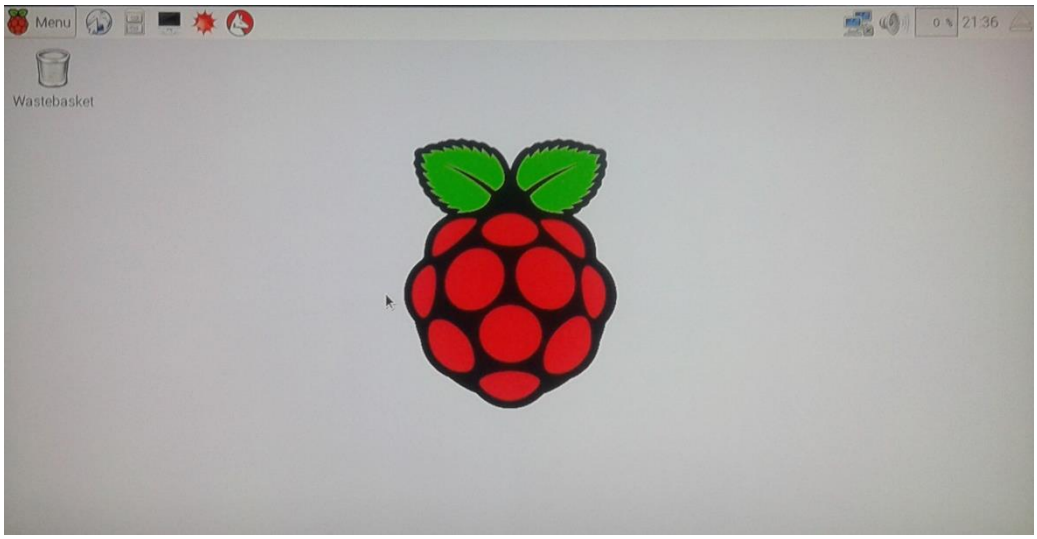


Fig.32: Escritorio de Raspbian.

Se abre una terminal y se ejecuta la siguiente línea de comando para buscar las actualizaciones del sistema operativo Raspbian que se acaba de instalar.

```
Sudo apt-get update.
```

Una vez encontrado los paquetes de actualización sobre la misma terminal se ejecuta la siguiente línea de comando para descargar e instalar los paquetes.

```
Sudo apt-get upgrade
```

Se debe recalcar que estos dos pasos de actualización y de instalación de los nuevos paquetes, se lo pueden realizar siempre y cuando se esté conectado a Internet, ya sea por medio de cable o wifi.

3.2 Clúster con RaspberryPi2

Iniciando en 2013, RaspberryPi2 (RPI2) se ha desarrollado para correr sistemas operativos ligeros, razón por la cual se ha acogido entusiastamente por la comunidad de Linux. Esta placa de desarrollo investigativo ha sido parte de la utilización del sistemas operativo Raspbian GNU/Linux y otros que se ejecutan en procesadores ARM. Para el desarrollo del clúster con RaspberryPi2, se debe tener conocimiento de los siguientes puntos:

- 3.2.1 Clúster servidor RaspberryPi2
- 3.2.2 Clúster servidor RaspberryPi2, características y esquema

3.2.1 Clúster servidor RaspberryPi2

El proyecto de desarrollo de clúster con RaspberryPi se originó por la necesidad de investigadores como Dennis Andrew en el estudio de sensores inalámbricos, para facilitar el uso de redes de datos de sensores. Dicho proceso de investigación hizo evidente el estudio de nuevas prácticas dentro de las redes, llevando al surgimiento de la creación de simulaciones distribuidas. En lugar de desarrollar todo el hardware de una sola vez, con este nuevo surgimiento de conocimientos y necesidades que se fueron presentando en el trascurso de la investigación se desarrolla una práctica llamada

simulación distribuida “En la que cada nodo de simulación se comportaría como un nodo sensor inalámbrico” (Dennis, Raspberry Pi Super Cluster , 2013), y por lo tanto éstos interactúan con todos los nodos que forman la red de simulación dentro de la LAN.

Un clúster Beowulf es un sistema de cómputo paralelo basado en clúster de ordenadores, ya sean personales o de escritorio, conectados en una red estándar sin el uso de otro equipamiento especializado para la computación paralela. Por lo general, el software de programación paralela es MPI (Message Passing Interface) utiliza TCP/IP, junto con algunas bibliotecas de desarrollo, que permiten a los programadores crear sus propios programas paralelos, y con ello poder dividir las tareas en segmentos más adecuados para manejarlos de una manera eficiente, y así ponerlos a funcionar dentro de varios equipos simultáneamente. El MPI asigna una API (Application Programming Interface) que logra una interacción de procesos asíncronos y síncronos, es por ello que el entorno del clúster dentro de esta investigación, proporciona una plataforma de desarrollo adecuada para el proyecto. (Dennis, Raspberry Pi Super Cluster , 2013)

El RPiCluster ofrece el desarrollo de nuevos proyectos que requieren de un MPI o Java en el entorno del Clúster, además el RPI proporciona una característica única de la placa, que es la interfaz de hardware de bajo nivel para la utilización de sistemas integrados, tales como I2C, SPI, UART y GPIO. Esto es de vital importancia para realizar el conjunto de pruebas de hardware integrado a gran escala. Como otra característica de la placa, cuando el usuario solo tiene acceso a un grupo del clúster este estaría bien siempre y cuando tenga todas las herramientas necesarias instaladas, caso contrario el usuario deberá trabajar con el administrador del clúster para realizar alguna tarea específica. Esto quiere decir que se requiere de la construcción de un grupo de trabajo personalizado, equipado de acuerdo a las necesidades particulares.

Estas bondades que ofrece RPI para la creación de un Raspberrypi clúster de bajo costo, conlleva un conjunto de inconvenientes potenciales. El mayor inconveniente podría ser que el RPI no es tan potente como una PC x86 actual. El RPI entre sus características de diseño cuenta con: ARM1176 de un solo núcleo procesador o (ARMv7), que funciona a 700MHz. Adicionalmente, puesto que el RPI utiliza un procesador ARM, que tiene una arquitectura diferente a los PC, es decir, ARM vs x86, cualquier programa MPI creado originalmente para x86 debe volverse a compilar

cuando se arranque RPI-Clúster. Este problema ventajosamente no está presente para los lenguajes de programación Java, Python y Perl. Finalmente por su capacidad limitada, el RPI-Clúster no soporta que múltiples usuarios accedan simultáneamente al sistema; es necesario crear un sistema de tiempo compartido que ayude a solucionar esta debilidad. (Dennis, Raspberry Pi Super Cluster , 2013)

3.2.2 Clúster servidor RaspberryPi2, características y esquema

Una de las características más importantes del estudio de RaspberryPi2 es la creación de un clúster con varias placas RaspberryPi. En este proyecto de investigación se creará dicho clúster con tres placas o nodos: dos serán nodos esclavos y uno principal, también se ocupará un *switch* Ethernet, al que se conectará mediante cable de red las máquinas de prueba. Todos los materiales a utilizar serán colocados dentro de una infraestructura para que se muestre de la siguiente forma. (Dennis, Raspberry Pi Super Cluster , 2013)

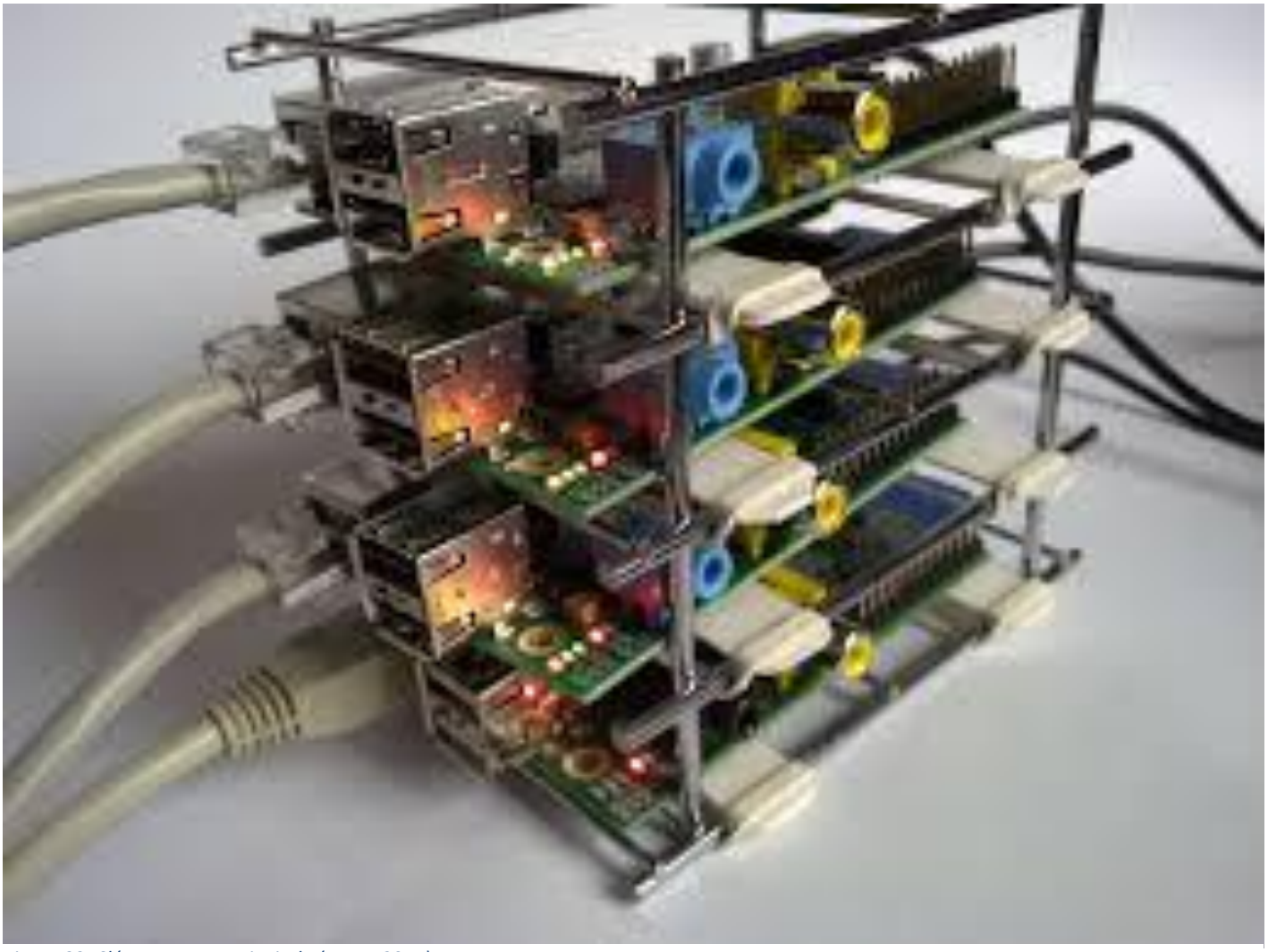


Figure 33: Clúster RasperryPi, Titulo (Dang, 2014).

Este sería el objetivo físico a alcanzar y el objetivo lógico o el esquema de conexión quedaría la siguiente manera. (Dennis, Raspberry Pi Super Cluster , 2013)

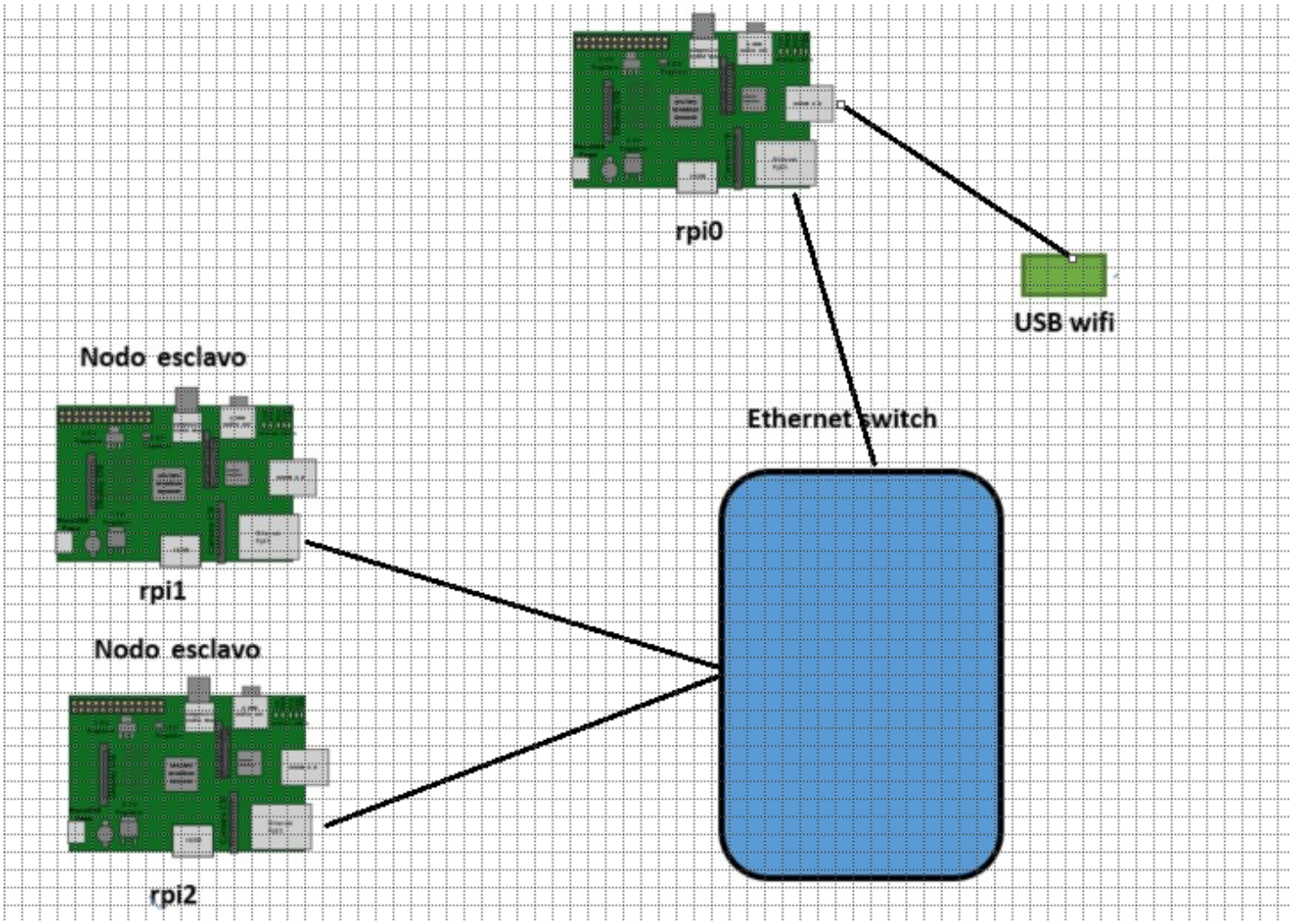


Fig.34: Modelo del Clúster RaspberryPi.

Este esquema, es un simple ciclo de conectividad de los RaspberryPi con los otros elementos, para que el clúster este estructurado en su forma física. Además se deberá instalar Raspbian, que se refiere a una versión del sistema operativo GNU/Linux y por lo tanto es open source, lo que facilita su adaptabilidad de acuerdo a las necesidades del usuario y al tipo de implementación. Este sistema operativo está basado en Debian Wheezy 7.0 específicamente para la placa de computadora SBC de RaspberryPi. El clúster que se va utilizar deberá utilizar una subred la más común es 192.168.2.201 para

su Ethernet 0 mientras que para el wireless del nodo maestro deberá ser una 192.168.1.1 asignada mediante un servidor DHCP, ya que tanto el wireless como la Ethernet deben ser diferentes. Por ello se utiliza el servidor DHCP que proporciona una configuración de red TCP/IP segura, y sobre todo que evita que existan conflictos por el ingreso manual de direcciones repetidas. Se utiliza un modelo cliente-servidor en donde el servidor DHCP mantiene un rango de direcciones IPs utilizadas en la red de una forma centralizada evitando estos conflictos. (Dennis, Raspberry Pi Super Cluster , 2013)

3.3 Configuración de la Red

La configuración de la red para la creación del clúster va a contar con:

- 1 Nodo Maestro (192.168.2.201)
- 2 Nonos Esclavos (192.168.2.202, 203 respectivamente)
- 1 Ip Virtual (192.168.2.205)
- 1 Ethernet Switch
- 1 USB wifi

Todo esto conectado mediante cables conectores utp, y de realizar las configuraciones adicionales en cada uno de los nodos, se logrará el esquema de conexión que muestra la Fig. 33. Cada una de estas configuraciones se indicará a continuación:

- 3.3.1 Configuración de la Red del Nodo Maestro
- 3.3.2 Configuración de la Red del Nodo Esclavos

3.3.1 Configuración de la Red del Nodo Maestro

Se empieza por la configuración del Nodo maestro, por lo que va a ser el equipo que controle a los otros periféricos. Un nodo maestro es designado mediante software para que sirva de coordinador de ciertas operaciones a los nodos esclavos en un clúster.

Para comenzar con la configuración del RaspberryPi, que servirá como nodo maestro, se debe editar el fichero, para ello se abre una terminal, y se ejecuta la siguiente línea de comando:

```
sudo nano /etc/network/interfaces
```

Una vez ejecutada, se mostrará el fichero con la configuración por omisión del RaspberryPi. Una vez abierto se procede a copiar las líneas de comando tal y como se muestra en la figura a continuación.

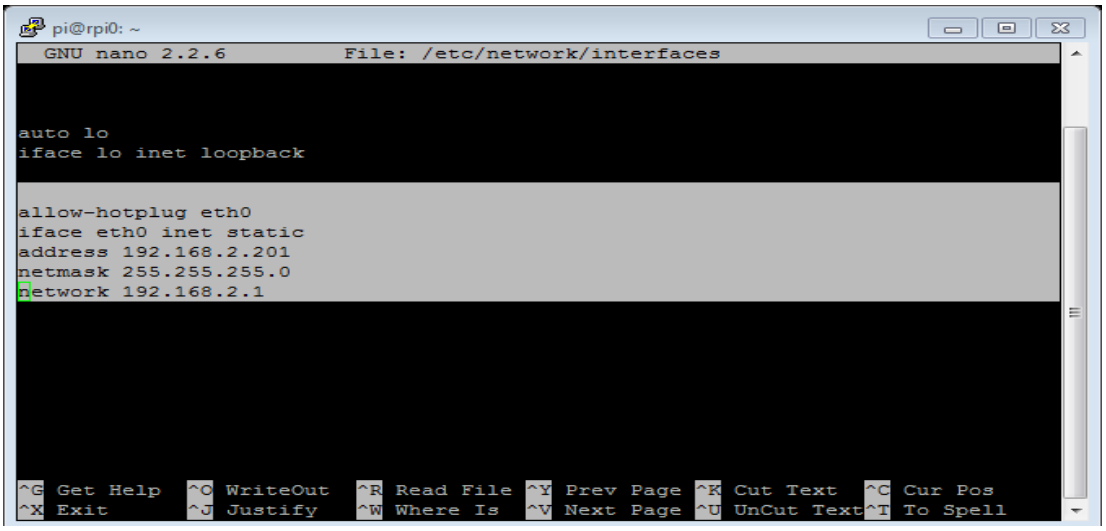


Fig.35: Configuración de la red en el nodo maestro, para tener acceso a internet.

Estas líneas que se agregan servirán para la configuración de la red ethernet (eth0), que es la que se utilizará, ya que se necesita hacer pruebas en distintos lugares. Para el wireless (wlan0) se necesita otra configuración con un nombre de red y una contraseña que se lo puede realizar siempre y cuando el clúster ya terminado esté en un lugar establecido, ya que se necesita de una red previamente implementada.

A continuación, se realizarán cambios en el fichero *sysctl.conf*, con el fin de que los nodos esclavos puedan acceder al internet por medio del nodo maestro que es por el que pasarán todas y cada uno de los permisos y peticiones a ser aprobadas o rechazadas. Se ejecuta la siguiente línea de comando:

```
sudo nano /etc/sysctl.conf
```

Luego se mostrará un fichero en donde se quita el comentario de la línea: *#net.ipv4.ip_forward=1*, esto se lo hace con la finalidad de compartir el acceso a internet de los nodos esclavos. Y quedará como se muestra en la figura.

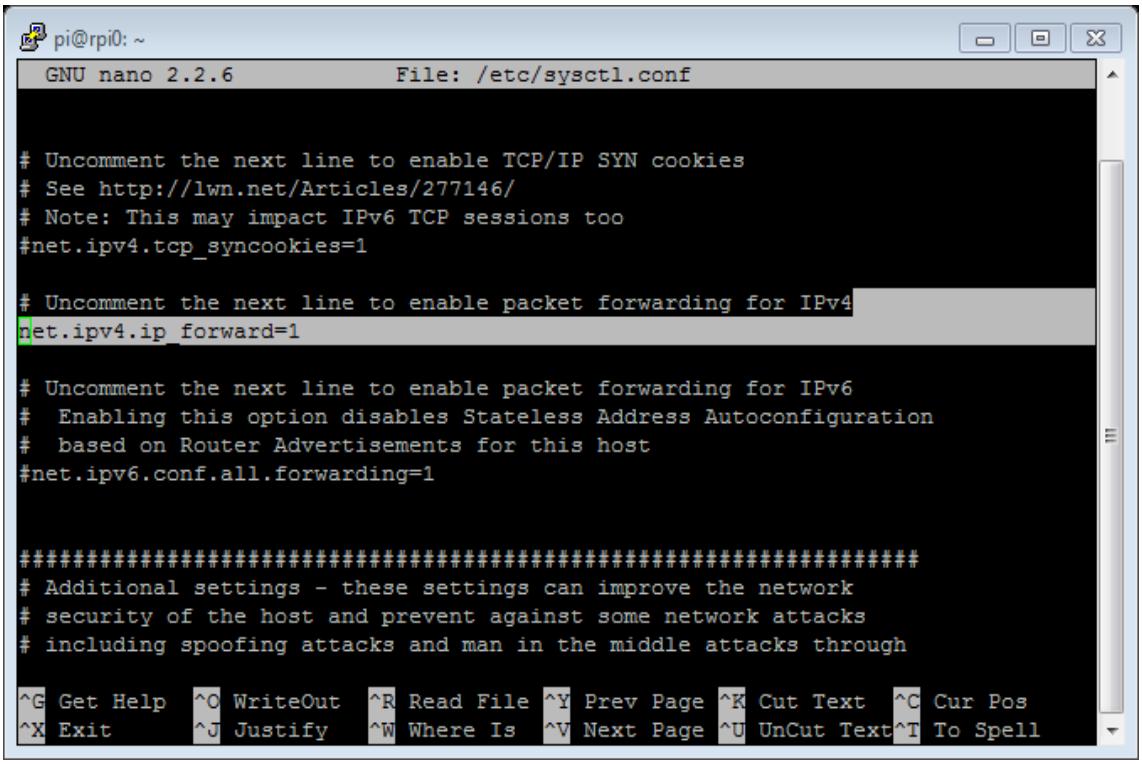


Fig.36: Configuración de la red en el nodo maestro, para acceder a internet a través del nodo maestro.

En el momento que se le quita el comentario de esa línea de comando, la IP Forward estará activa permanentemente, esto lo realizará en cada momento en que se reinicie el servidor.

Como paso siguiente se debe realizar cambios en el cortafuegos o firewall del sistema, para así tener comunicación entre nodos, y a su vez poder acceder a internet a través del nodo maestro, esto se da configurando el firewall de tal manera que en el momento que arranca el sistema este cambio se ejecute, para ello, se abre el siguiente fichero con la siguiente línea de comando: *sudo nano /etc/rc.local*, y en la parte inferior del mismo se procede agregar la línea marcada tal y como lo muestra la imagen:

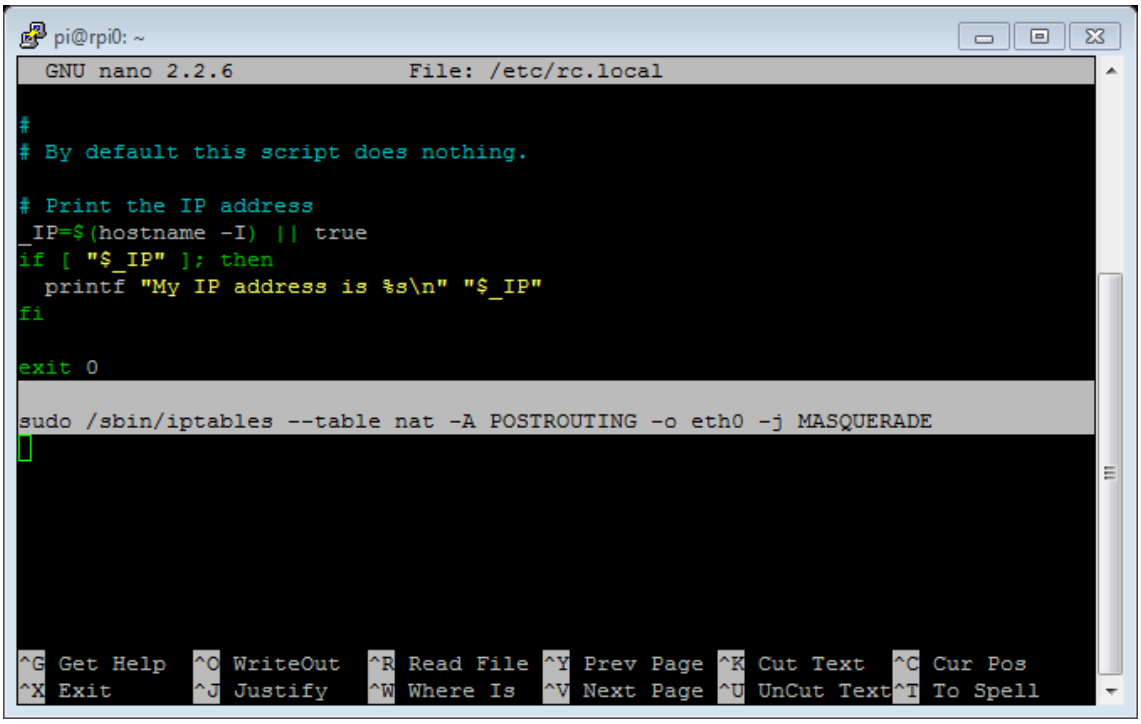


Fig.37: Cambios en el firewall.

Continuando con la configuración, es el momento de indicar que ips pueden ser vistas por los nodos y tener una comunicación entre ellos, para ello se ejecuta lo siguiente:

```
sudo nano /etc/hosts
```

Al ejecutar esta línea de comando se mostrará el fichero, y en la parte final se agrega todas las líneas de comando señaladas como se muestra en la figura.

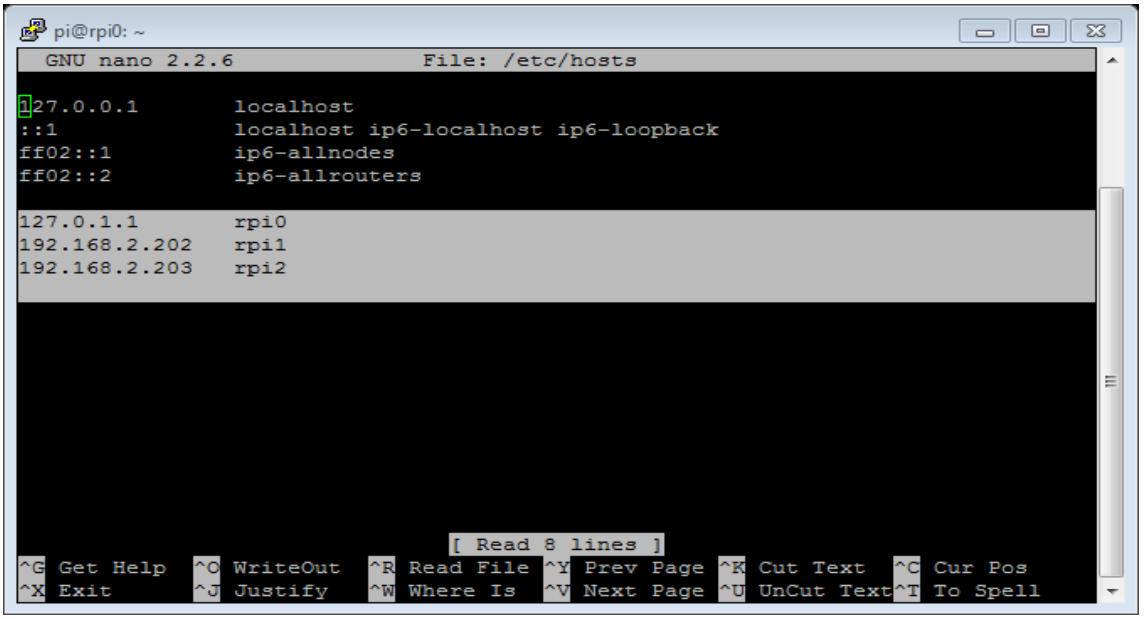


Fig.38: Configuración de la red en el nodo maestro, ingreso de ips de los nodos que estarán conectados en red.

Estas líneas de comando se deben ingresar para la comunicación entre los nodos esclavos y el nodo maestro. Por lo consiguiente cada uno de los nodos sabe a qué nodos puede acceder y con cuáles nodos pueden compartir información, para ello se agrega las siguientes líneas de comando indicadas en la imagen anterior, en cada uno de los nodos.

Este código variará tanto en IPs como en los nombres, esto dependerá del nodo que se esté configurando y también los nodos con los que se quiera compartir información o tener una comunicación. Una vez concluido todo esto, se debe reiniciar la máquina que se está configurando para que los cambios tomen efecto.

3.3.2 Configuración de la Red del Nodo Esclavos

La configuración de los nodos esclavos se enfocará en qué tipo de información se desea que estos transmitan, ya que por lo general no se transmite información sin antes hacer una petición al nodo maestro, y a su vez, estos nodos esclavos no pueden pasar ninguna información entre sí, es por ello, la necesidad de un nodo maestro, todos los nodos existentes en la red deberán hacer peticiones al nodo maestro de alguna información en particular. La importancia de un test de red radica en que el nodo maestro haga ping a cada nodo esclavo.

Para realizar lo antes descrito, se debe cambiar la configuración de algunos ficheros, estos cambios se los realizará en todos y cada uno de los nodos esclavos con los que esté conectado el clúster, así como también el nodo maestro. Para ello, se ejecutara las siguientes instrucciones.

```
sudo nano /etc/hostname
```

Luego se abre el fichero y se procede a renombrar a cada uno de los raspberryPi que llevan un nombre por omisión, este nombre puede ser cualquiera, esto se lo hace con el afán de identificarlos unívocamente, para el caso será **rpi0.**

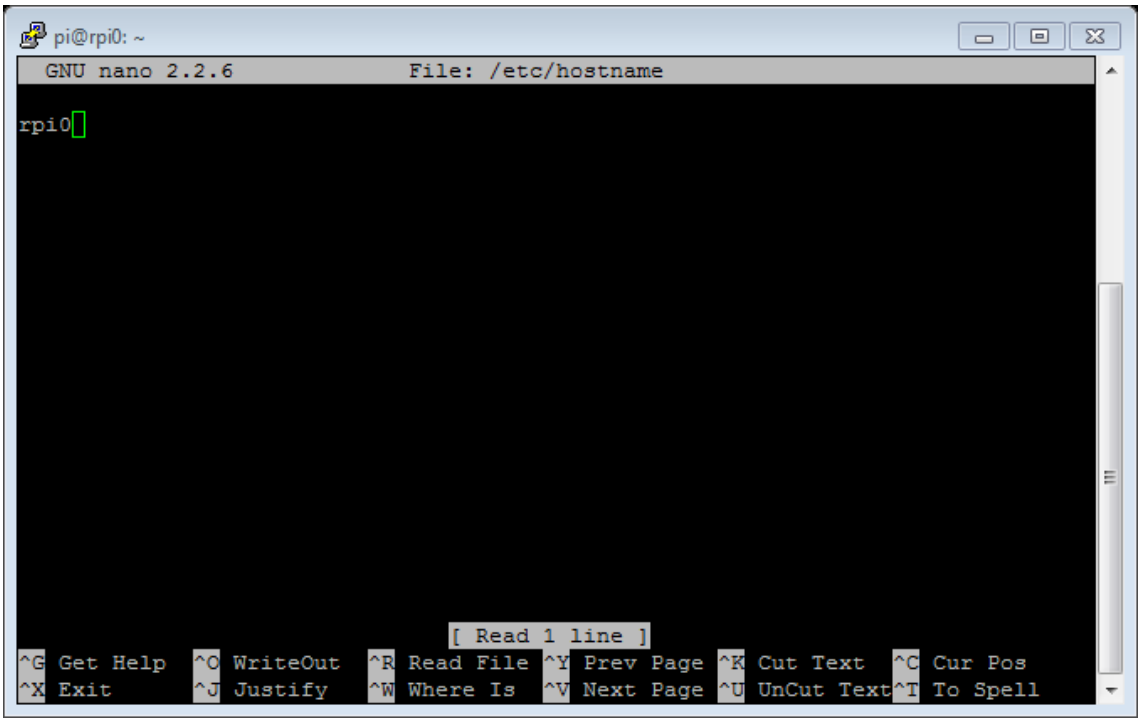


Fig. 39: Configuración de la Red del Nodo Esclavos, renombrar el dispositivo.

Se procede también a editar el archivo */etc/network/interfaces.*

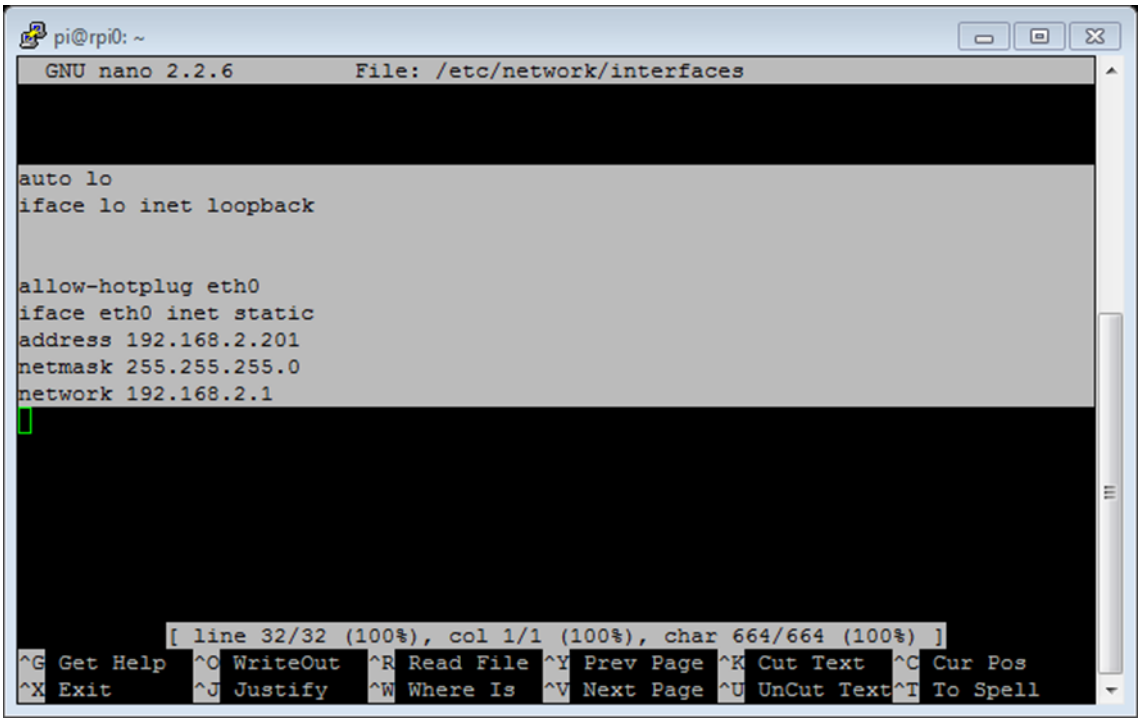


Fig. 40: Configuración de la Red del Nodo Esclavos, ingreso de líneas para tener comunicación entre nodos.

Esto se lo realizará para todos y cada uno de los nodos existentes tanto esclavos y maestro, y así crear un puente de comunicación entre ellos. Lo que variará en cada uno de ellos será su address que llevara la dirección ip que se le configuró con anterioridad en el nodo maestro.

Además se crea un directorio para poder ingresar en cada uno de los nodos sin tener que utilizar una contraseña, este directorio será */.ssh* y se lo crea de la siguiente manera.

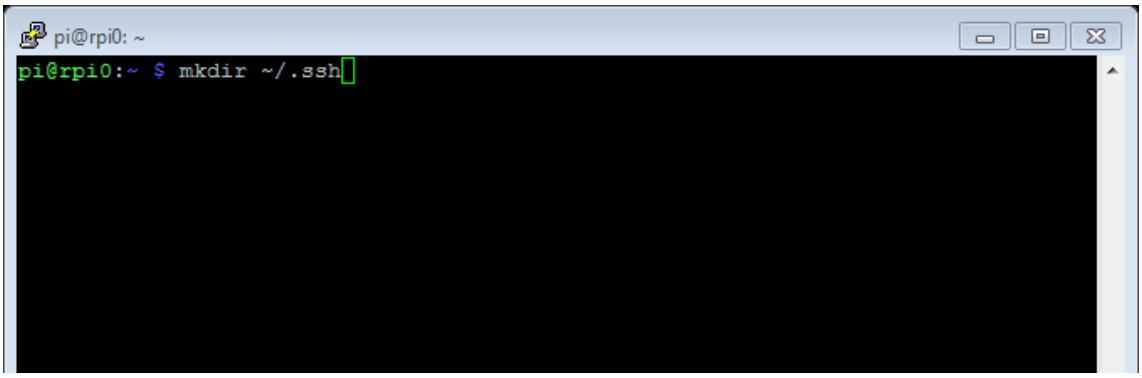


Fig.41: Configuración de la Red del Nodo Esclavo, creación de directorio .ssh para acceder a los nodos.

Se reinicia el equipo configurado y se verifica los cambios que han sido guardados. Una vez concluido, se procederá a realizar los mismos pasos para todos los nodos esclavos restantes.

Una vez concluido esto, para verificar la configuración realizada en los ficheros, y de que estos cambios realizados permitan comprobar la conexión de red. Se hace un ping desde el nodo maestro hacia los nodos esclavos de la siguiente manera.

```
sudo ping rpi1
```

Se ejecuta esta línea de comando para todos los nodos esclavos con los que cuenta el clúster y se verifica si existe respuesta, de ser así, se asegurará que la red funciona de forma correcta.

3.4 Conectar una USB en el nodo maestro

En esta etapa de la experimentación se montará la unidad de disco en el nodo maestro, y replicará esta operación en todos los nodos esclavos que formarán parte del clúster, esto servirá en lo posterior para implementar todos los servicios y carpetas.

Primero se instala *nfs-kernel-server* en el nodo maestro y *nfs-common* para cada uno de los nodos esclavos, estos sistemas de archivos permitirá acceder a archivos remotos a través de la red. A continuación se indica las configuraciones que se realizará en este tema:

- 3.4.1 Configuración de los nodo maestro para que el disco-USB se pueda replicar en cada uno de los nodos que forman parte del clúster.
- 3.4.2 Configuración de los nodos esclavos para que el disco-USB se pueda replicar en ellos

3.4.1 Configuración de los nodo maestro para que el disco-USB se pueda replicar en cada uno de los nodos que forman parte del clúster.

Se procede a instalar el *Kernel-Server* desde la terminal del nodo maestro.

```
sudo apt-get install rpcbind nfs-common nfs-kernel-server
```

Se edita el archivo */etc/exports* para montar en el nodo el dispositivo USB con los privilegios para que los nodos esclavos lo reconozcan.

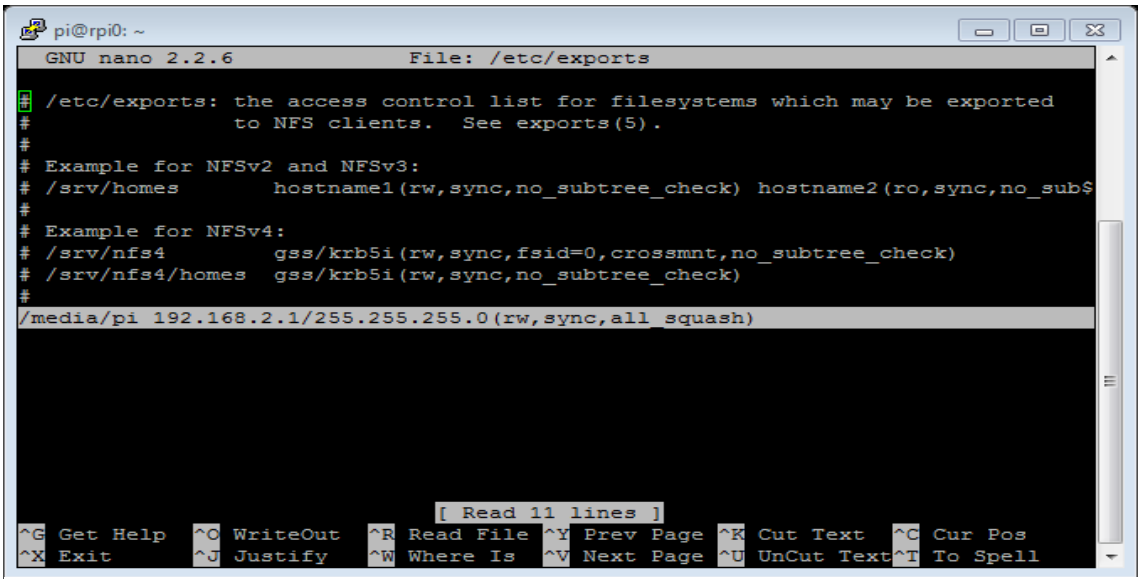


Fig. 42: Configuración de la USB en el nodo maestro, sistema de archivos exportados.

El archivo */etc/exports* hace un control de cuales sistemas de archivos son exportados a los nodos esclavos, cada sistema de archivos exportados debe insertar su propia línea como se muestra en la imagen anterior y la lista de hosts colocada después del sistema de archivos a exportar.

Así mismo, es necesario modificar el archivo */etc/fstab*, en donde se le indica al sistema operativo la partición y la carpeta en donde debe montarse para que sea identificada en los nodos esclavos.

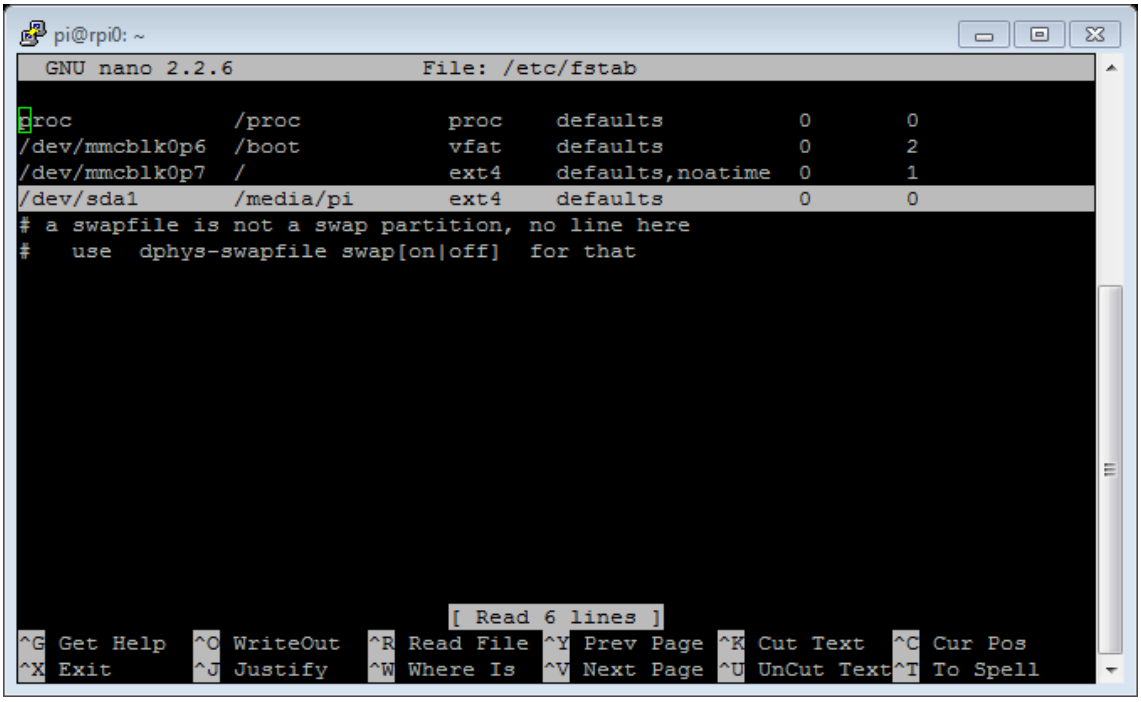


Fig.43: Configuración de la USB en el nodo maestro, lista de discos y particiones disponibles.

Se guarda los cambios en el fichero y se procede a ejecutar la siguiente línea de comando.

```
Sudo nano /etc/hosts.deny
```

Y se inserta la línea que se encuentra señalada en la gráfica. Esta línea se la debe insertar en la parte inferior del fichero.

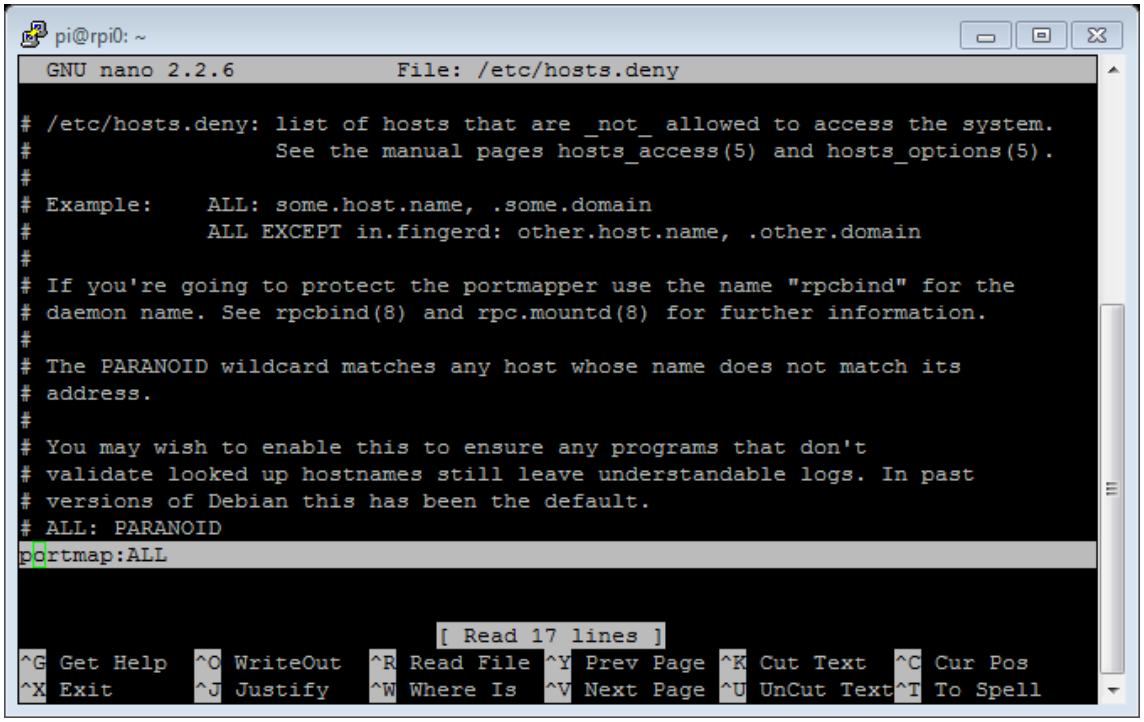


Fig. 44: Configuración de la USB en el nodo maestro, archivo de acceso al servidor.

Al insertar esta línea *portmap:All* se encarga de gestionar la asignación entre aplicaciones y puertos, es decir el portmap es el mapeador de Linux. Una vez hecho esto, se procede a guardar los cambios y a salir del fichero.

Una vez realizado este paso, se ejecuta la siguiente línea de comando.

```
Sudo nano /etc/hosts.allow
```

Y se mostrará el fichero en donde se ingresa las líneas señaladas.

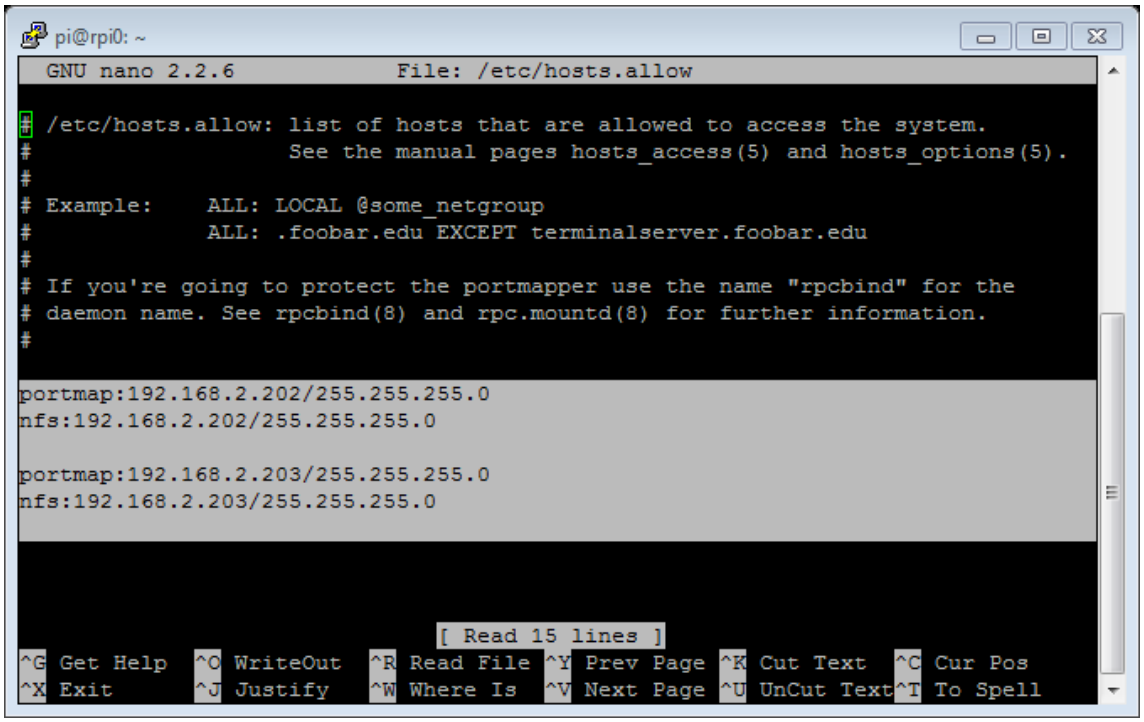


Fig. 45: Configuración de la USB en el nodo maestro, archivos de acceso.

Se realiza esto con la finalidad de que exista una comunicación entre el nodo maestro y los nodos esclavos, y así poder levantar el servicio *nfs* en los nodos esclavos, para que estos puedan visualizar la unidad USB previamente instalada en el nodo maestro. Se guarda los cambios y se procede a reiniciar los servicios con las siguientes líneas de comando.

```
sudo /etc/init.d/nfs-common restart
```

```
sudo /etc/init.d/nfs-kernel-server restart
```

Se ejecutara cada una de las líneas de comando y posteriormente se procede a verificar su estado en cada una de ellos, para así descartar cualquier error posible. Esto se lo hace con las siguientes líneas de comando.

```
sudo /etc/init.d/nfs-common status
```

```
sudo /etc/init.d/nfs-kernel-server status
```

Una vez ejecutada cada una de estas líneas, mostrará una imagen como la que se muestra a continuación. Indicando los posible fallos o su normal funcionamiento.

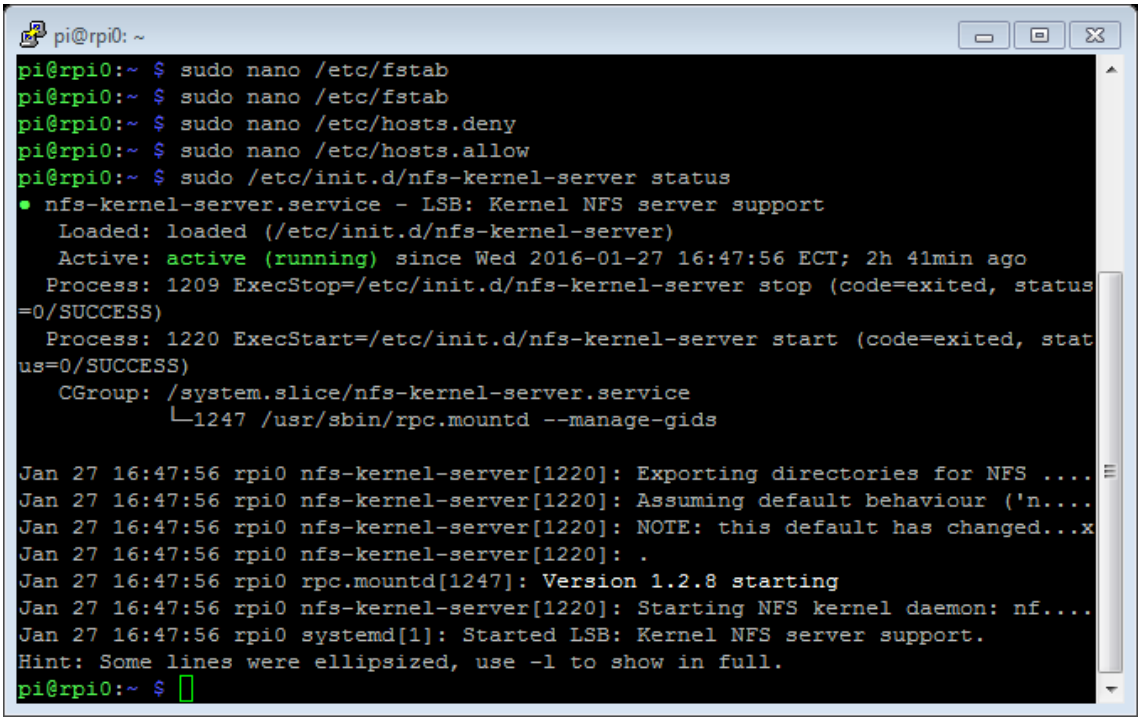


Fig. 46: Configuración de la USB en el nodo maestro, estado del nfs-kernel-server .

Si no existió ningún error dentro de los pasos antes mencionados, la configuración del nodo maestro queda finalizada.

3.4.2 Configuración de los nodos esclavos para que el disco-USB se pueda replicar en ellos

Se abre una terminal de cualquier nodo esclavo y se procede a instalar el Kernel para el cliente, desde la terminal se ejecuta la siguiente línea de comando.

```
sudo apt-get install rpcbind nfs-common nfs-kernel-server
```

Una vez ejecutada, aparecerá la siguiente pantalla mostrando la instalación de los paquetes necesarios para la configuración del nodo.

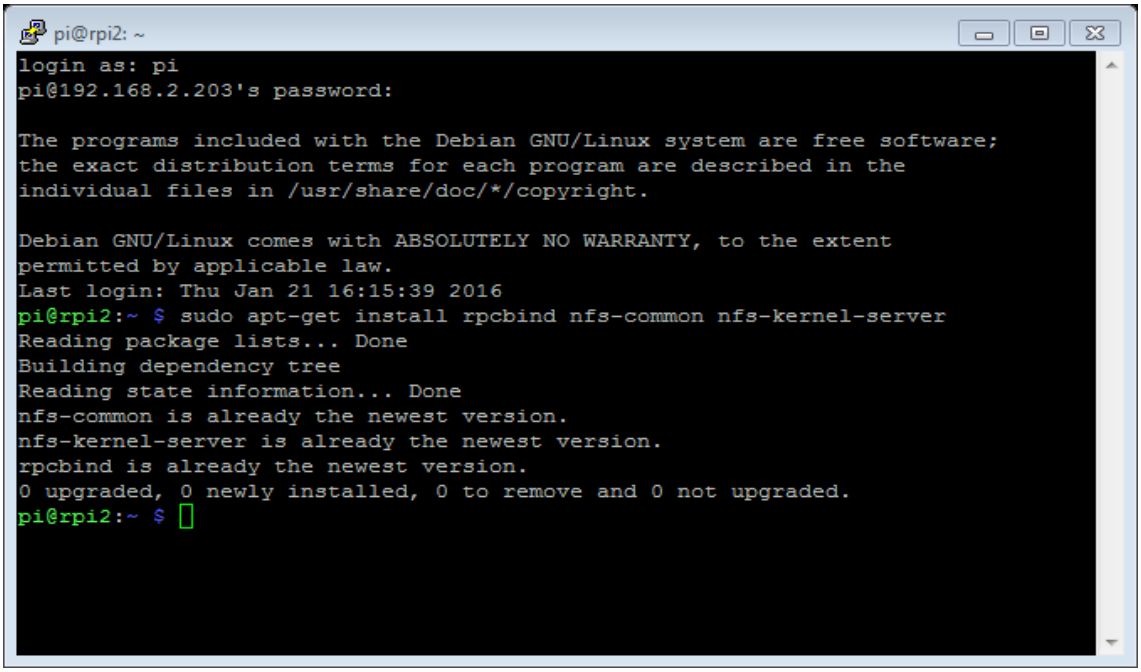


Fig. 47: Configuración de la USB en el nodo esclavo, instalación del nfs-kernel-server nodo esclavo.

Concluida la instalación de los paquetes en el nodo se debe ejecutar la siguiente línea de comando para abrir el fichero *fstab.*

```
sudo nano /etc/fstab
```

Una vez posesionados sobre el fichero se agrega la siguiente línea de comando que se muestra en la imagen.

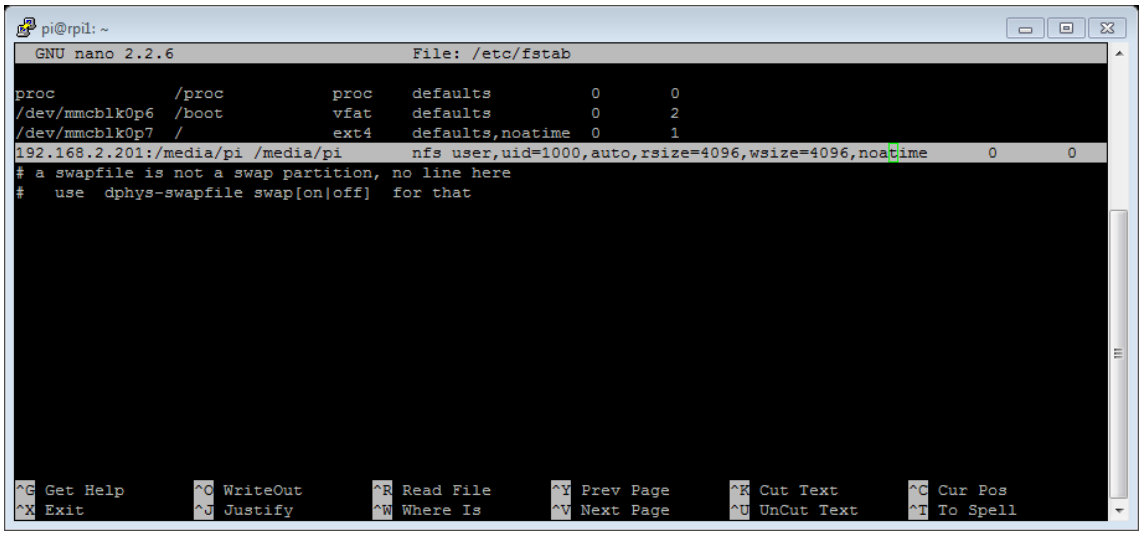


Fig. 48: Configuración de la USB en el nodo esclavo, lista de discos y particiones disponibles.

Este fichero muestra las particiones que existen actualmente, y la línea que se acaba de ingresar indica el servidor NFS y nombre del directorio a montar, el punto de montaje local, el tipo de sistema de archivos, y las opciones de montaje.

Una vez realizado esto se ejecuta la siguiente línea de comando para reiniciar el dispositivo.

```
sudo reboot
```

Se realiza este paso, con la finalidad de que los cambios realizados se ejecuten y corran de la manera correcta. Esto de reiniciar el equipo se lo realiza cada vez que se haga algún cambio sobre cualquier fichero o sobre cualquier nodo. Luego de ello se ejecuta la siguiente línea de código.

```
sudo /etc/init.d/nfs-common status
```

En el momento que se ejecuta esta línea de comando, mostrará una imagen como la que se indica a continuación, donde aparece el estado del nfs y qué servicios se encuentran corriendo actualmente, además de los errores que en ellos se han encontrado.

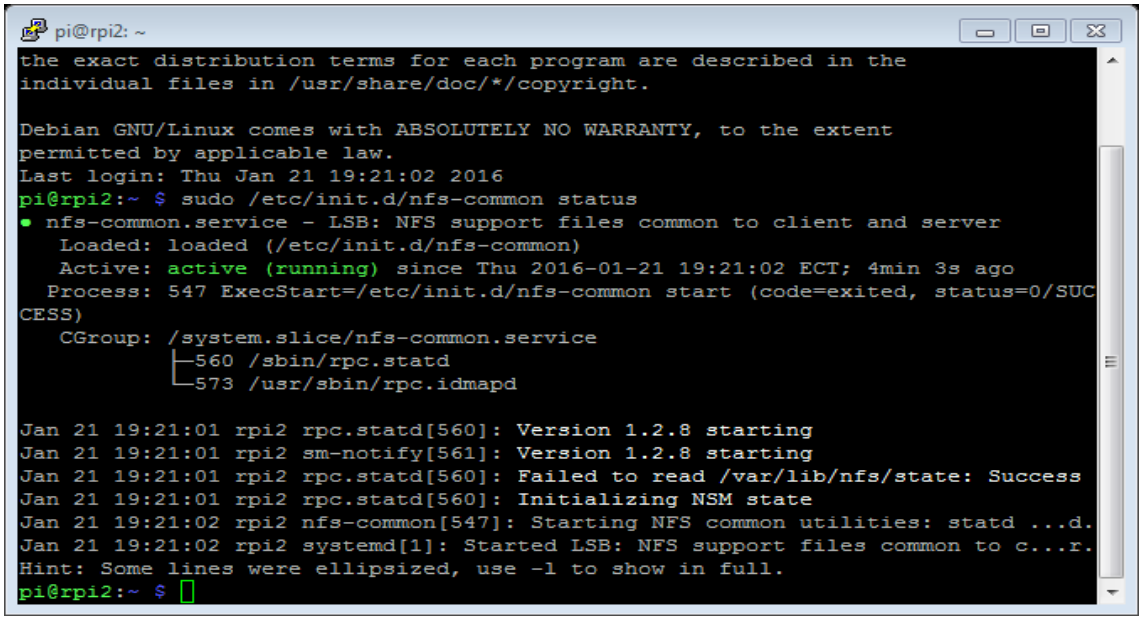


Fig. 49: Configuración de la USB en el nodo esclavo, estado nfs-common.

La imagen anterior muestra un error:

Failed to read /var/nfs/satate: Success

Si este error se presenta, es porque una librería del nfs no se está ejecutando de manera correcta o simplemente no está corriendo. Para ello se debe ejecutar la siguiente línea de código.

```
sudo /etc/init.d/nfs-common restart
```

Una vez ejecutada la siguiente línea de comando, mostrará el archivo *nfs* sin ningún error como lo indica la imagen que se muestra a continuación, ejecutando nuevamente la siguiente línea de comando.

```
sudo /etc/init.d/nfs-common status
```

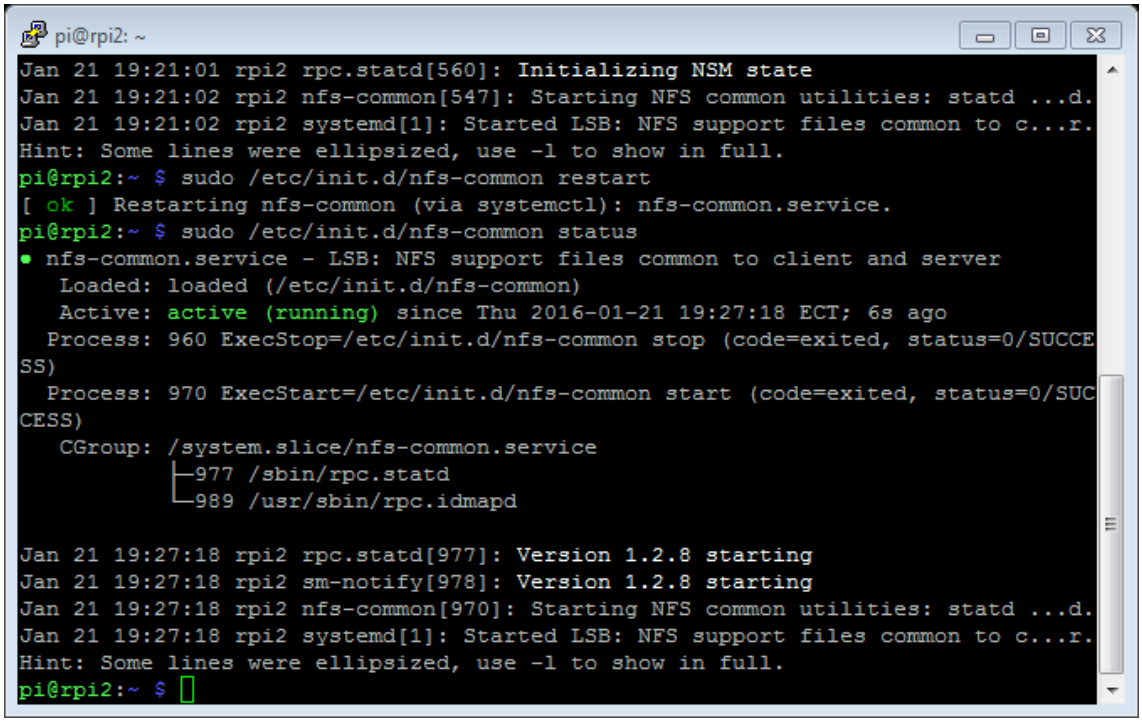


Fig. 50: Configuración de la USB en el nodo esclavo, estado nfs-common.

Con esto se podrá asegurar de que todos los servicios del nfs están corriendo correctamente. El siguiente paso será ejecutar la siguiente line de comando

```
sudo /etc/init.d/rpcbind status
```

Mostrará una imagen igual a la que se indica a continuación, sin ningún error. En caso de haber algún error se deberá proceder como se lo hizo en los pasos anteriores cambiando el **status** por **restart**.

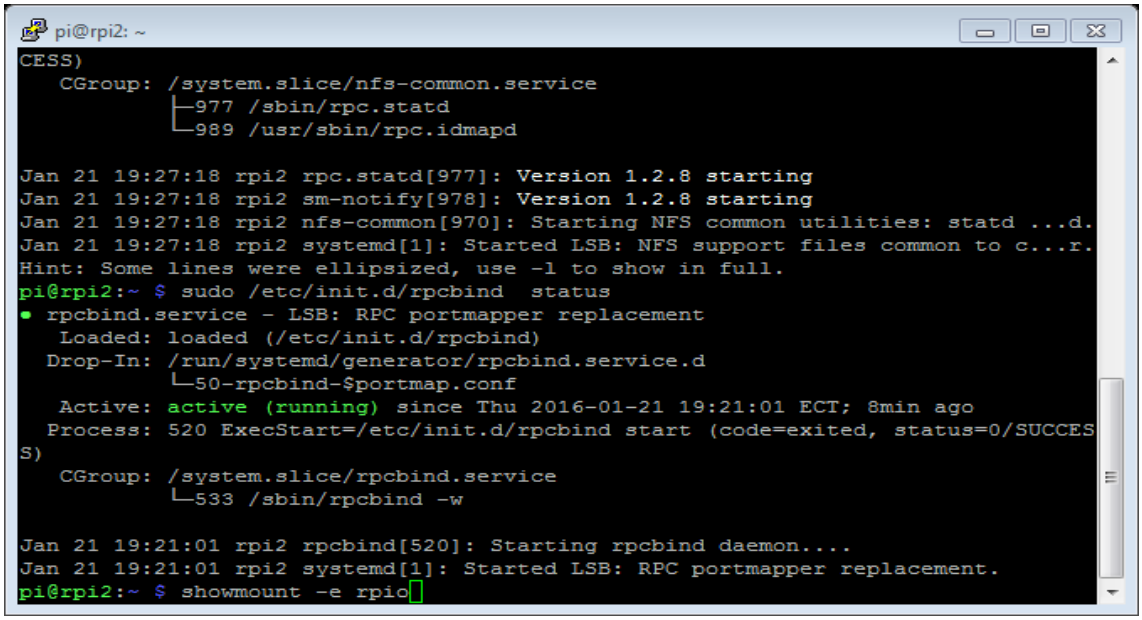


Fig. 51: Configuración de la USB en el nodo esclavo, estado rpcbind .

Se ejecuta la siguiente línea de comando para poder visualizar la unidad montada sobre el nodo maestro en el nodo esclavo que se esta configurando.

```
showmount –e  rpi0
```

Mostrará la siguiente imagen.

Fig. 52: Configuración de la USB en el nodo esclavo, visualización de la unidad montada.

Una vez mostrada la imagen se ejecuta esta línes de codigo, dicha línea de codigo indica que se monta correctamente el directorio */media/pi* en el nodo que se esta trabajando bajo el mismo directorio */media/pi.*

```
sudo mount –t nfs –o rw rpi0:/media/pi /media/pi
```

Se procede a ejecutar la siguiente línea de código

```
df –h
```

Esto se lo debe realizar para visualizar si se ha montado la USB sobre el dispositivo en el que se ha estado trabajando. Si toda la configuración esta correcta, mostrará la siguiente imagen con la USB ya montada sobre nuestro nodo esclavo.

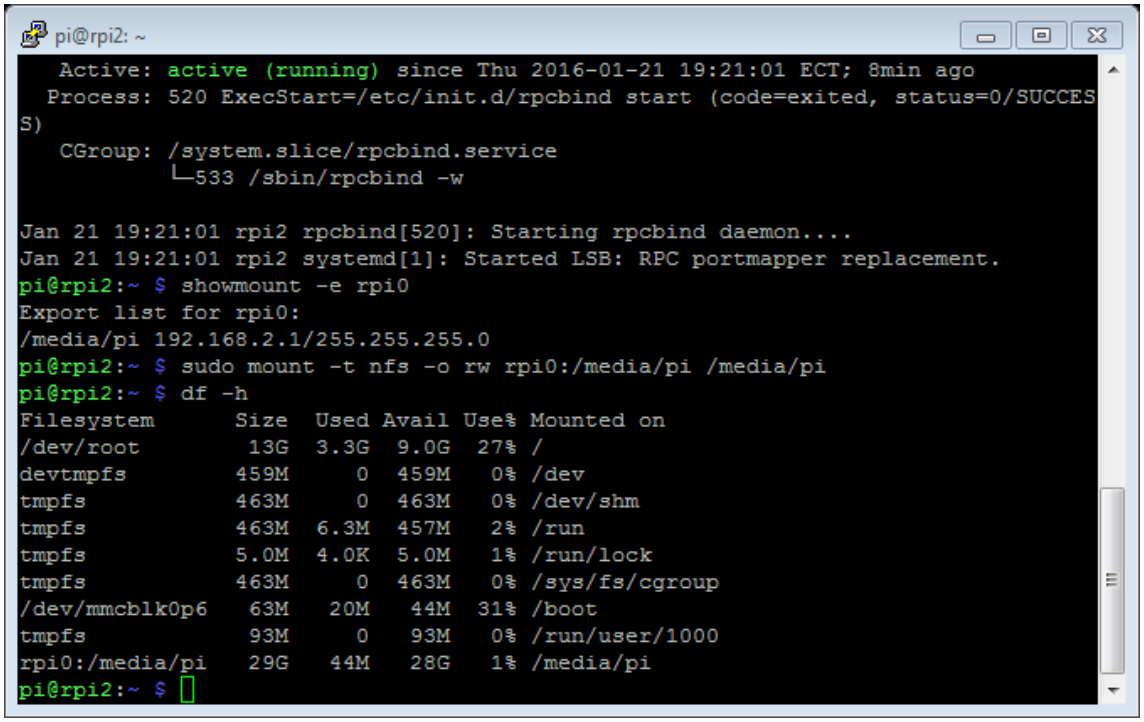


Fig. 53: Configuración de la USB en el nodo esclavo, muestra las unidades montadas.

3.5 Configuración del Servidor FTP

La configuración del archivo FTP, se lo realiza con la finalidad de poder compartir archivos fuera de la red local. Es decir desde la Internet, desde cualquier lugar donde se tenga acceso a este servicio. Para ello no hay nada mejor que la instalación del servidor FTP. En el sistema operativo Raspbian se puede instalar varios servidores FTP, pero para el caso de estudio, se utilizará *proftpd*, por su flexibilidad y al fácil manejo de la configuración de carpetas que se quiera compartir en una red. Los subtemas que describirán el proceso de instalación del servidor, creación de carpetas, y accesos a los usuarios, serán los siguientes:

- 3.5.1 Instalación del Servidor FTP
- 3.5.2 Creación de Carpetas Compartidas
- 3.5.3 Acceso a los Usuarios

3.5.1 Instalación del Servidor FTP

Lo primero que se debe realizar es instalar el servidor en cualquier nodo esclavo. Y se procede a ejecutar la siguiente línea de código:

```
sudo apt-get install proftpd
```

Al momento de ejecutar esta línea, aparecerá una pantalla como la siguiente:

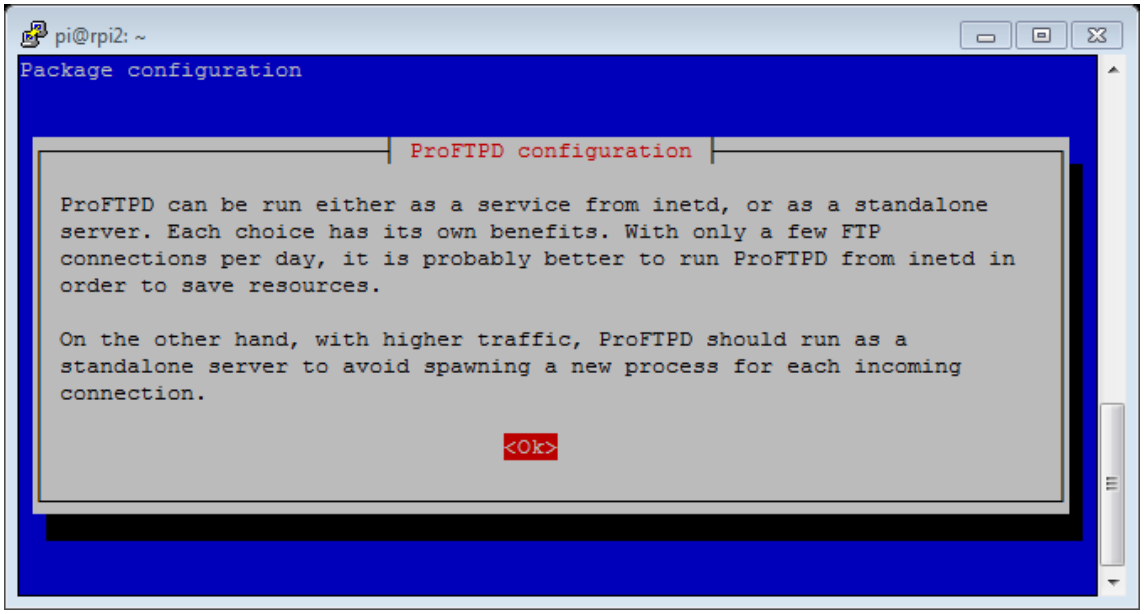


Fig. 54: Instalación del servidor FTP.

Esta pantalla indica la configuración y en donde se están instalando los paquetes. Se da clic en ok para continuar la instalación

Durante el proceso de instalación se mostrará una ventana, en la cual se pregunta qué tipo de servidor se quiere instalar. Para el caso de estudio, se deja el que se instala por omisión:(standalone).

3.5.2 Creación de Carpetas Compartidas

Una vez finalizado el proceso anterior, se inicia con la creación de carpetas en las que se van a guardar los archivos. Para ello, es necesario que el paso 3.3 y los relacionados, se hayan ejecutado de forma correcta, sin problemas. Se creará en la USB, que está previamente montada, una carpeta o las carpetas que se vayan a utilizar sobre el directorio */media/pi*, llamada ftp y dentro de ella las carpetas cargar y descargar. Esto, se lo realiza para que los usuarios puedan enviar archivos al servidor. Para ello, se escribirá el siguiente código:

```
cd /media/pi
sudo mkdir ftp
cd ftp
sudo mkdir cargar
sudo mkdir descargar
sudo chmod 777 cargar
sudo chmod 777 descargar
```

Las dos últimas líneas se las ejecuta para dar todos los permisos a las carpetas, y que estas no presenten problemas a la hora de crear y acceder a las carpetas desde otro nodo. Al concluir con el proceso, los usuarios a los cuales se les haya otorgado los permisos necesarios podrán acceder a las carpetas compartidas.

3.5.3 Acceso a los Usuarios

Al momento en que se realiza la instalación del *proftpd*, se crea por defecto un usuario llamado **ftp**, este usuario es el medio por el cual se podrá acceder al servidor, y el cual tendrá acceso a las carpetas que ya estén creadas con anterioridad. Para ello, se edita el fichero */etc/passswd*, y se incluye en la parte inferior la línea de código como se indica a continuación.

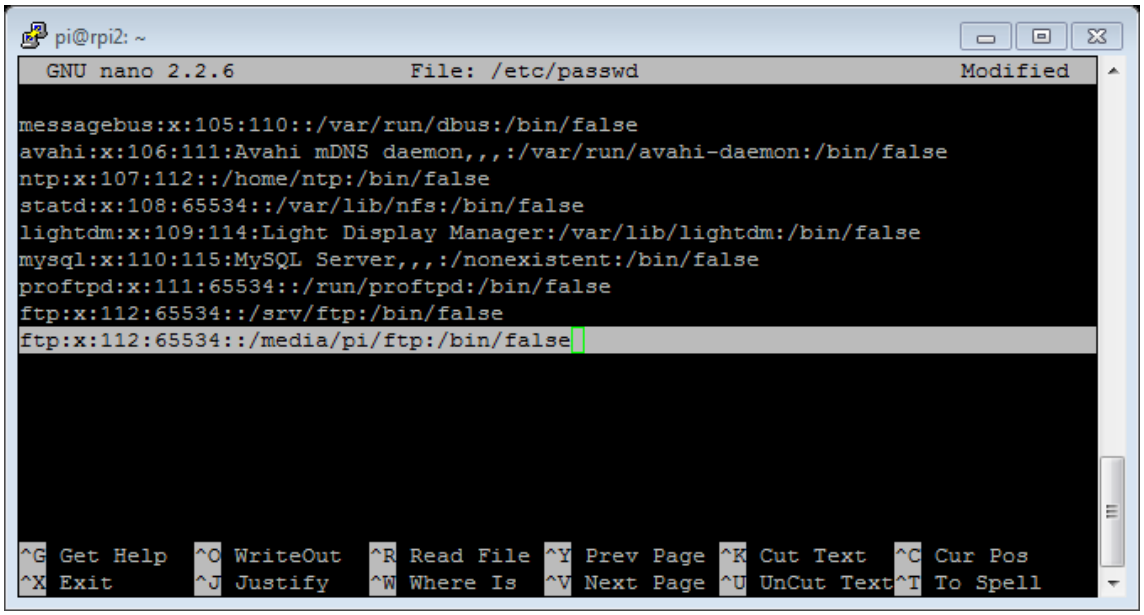


Fig. 55: Acceso a los Usuarios FTP.

Con esto se comprueba que el usuario corresponda con el directorio */media/pi/ftp* y un intérprete de comandos */bin/false*. Este cambio conlleva modificar el fichero */etc/shells*, insertando una línea con el código */bin/false* en su parte inferior, para que exista una interpretación de los comandos tal y como se muestra a continuación:

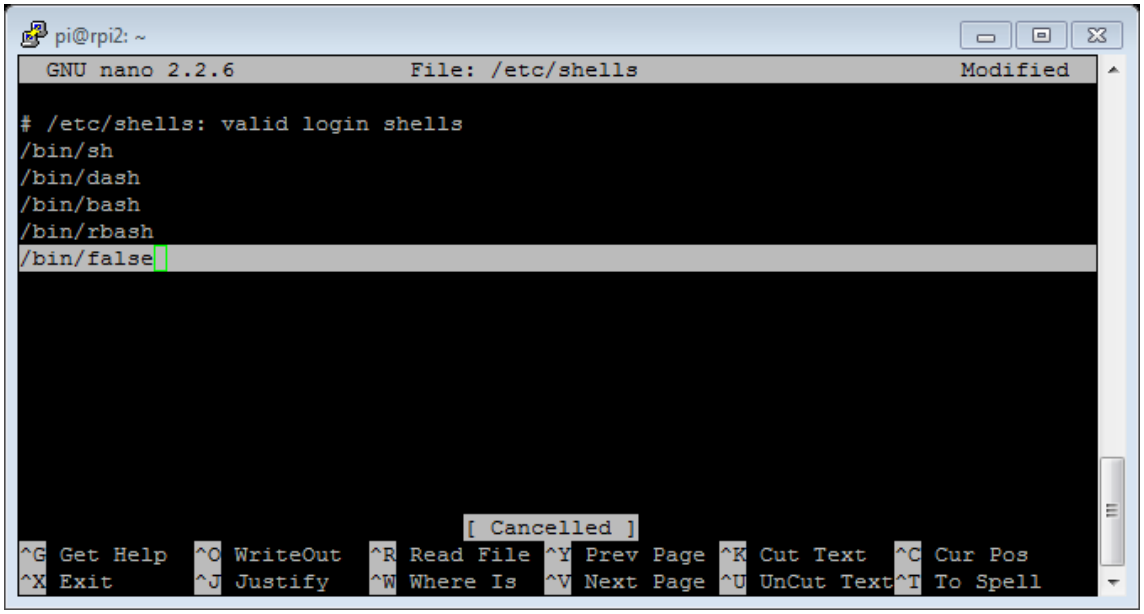


Fig. 56: Acceso a los Usuarios FTP, interprete de comandos.

Lo que queda pendiente es establecer una contraseña al usuario ftp, como un medio de seguridad; se ingresa la contraseña dos veces, ya que requiere ser confirmada. Para ello se ejecuta el comando:

```
sudo nano passwd ftp
```

Es necesario configurar adecuadamente el servidor, por lo que se edita el fichero */etc/proftpd/proftpd.conf*, pero previo a ello, se realiza un respaldo del fichero ejecutando la siguiente línea de comando:

```
sudo cp /etc/proftpd/proftpd.conf  /etc/proftpd/proftpd.conf.original
```

Una vez que se ha obtenido una copia del fichero proftpd.conf, se puede configurar el servidor. Pero antes se debe ubicar en el mismo directorio donde se encuentra l *proftpd*, es por ello que ejecuta lo siguiente:

```
cd /etc/proftpd
```

Ya con la seguridad de estar sobre el directorio del *proftpd*, se ejecuta lo siguiente

```
sudo nano proftpd.conf
```

Una vez en el fichero se procede a agregar las siguientes líneas de código, resaltadas en la imagen:

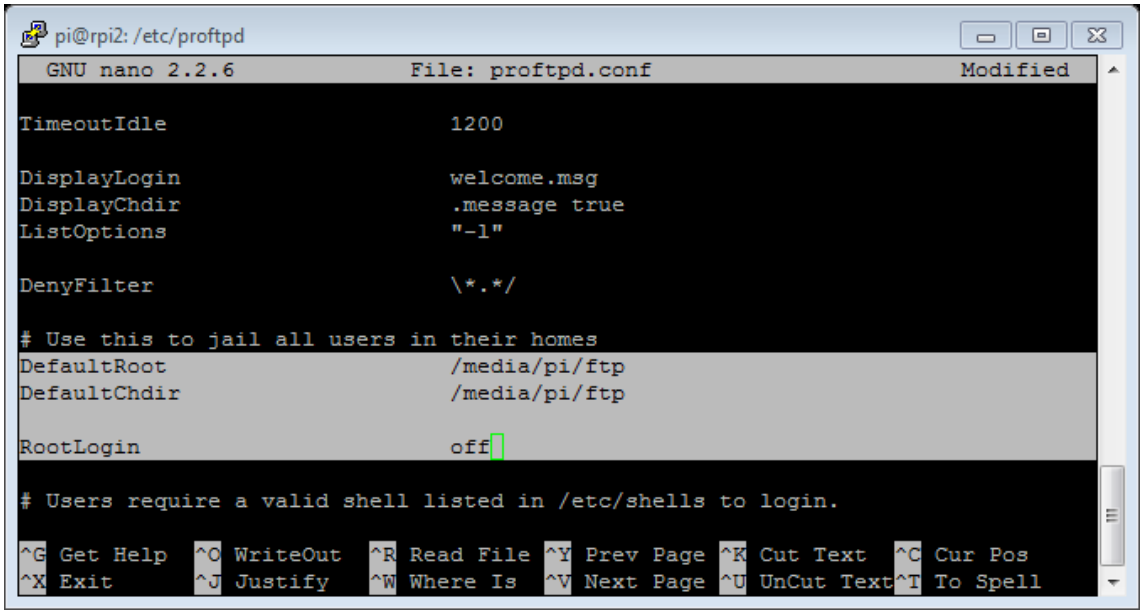


Fig. 57: fichero de verificación de usuarios.

Las primeras dos líneas provocan que el usuario quede registrado en la carpeta que se ha creado anteriormente, y con ello tener un árbol de directorios. La línea *RootLogin off* impedirá que cualquier usuario pueda acceder al servidor como root.

En el mismo fichero se modifica la línea *PassivePorts 25010 25090*, que se refiere al número de puertos pasivos que usará el servidor.

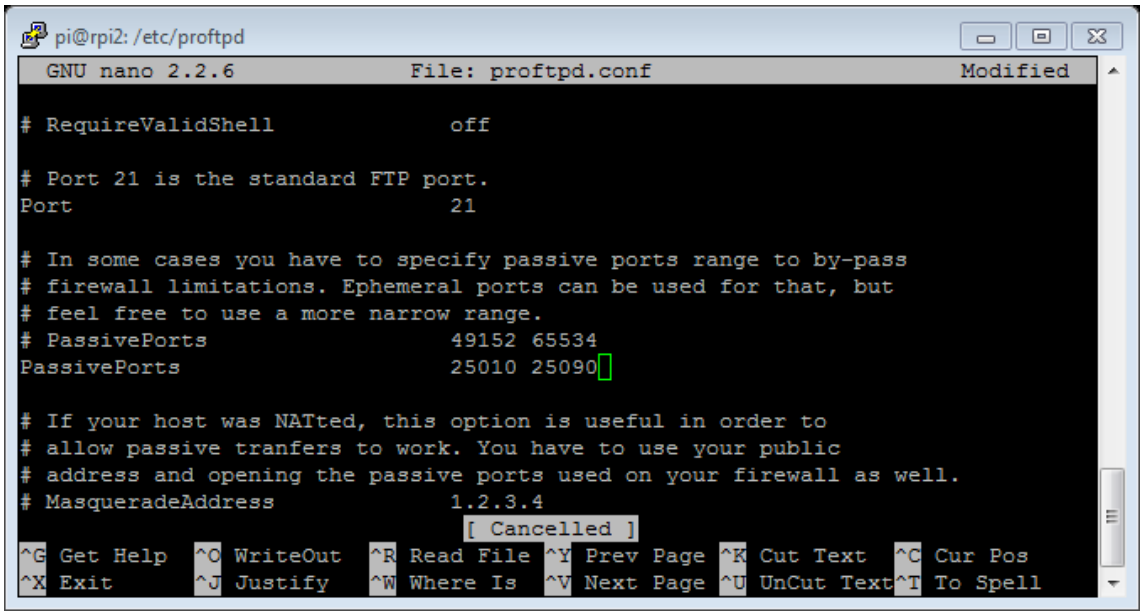


Fig. 58: configuración de numero de puertos que se utilizará servidor.

También, se debe ingresar las líneas que se encuentran señaladas en la figura. Estas líneas de comando habilitan o deshabilitan: ficheros sobrescritos, descargas ininterrumpidas, subidas ininterrumpidas, y eliminar las subidas suspendidas. Esto dependerá del estado en el que se encuentren cada una de ellas, este estado puede ser encendido o apagado

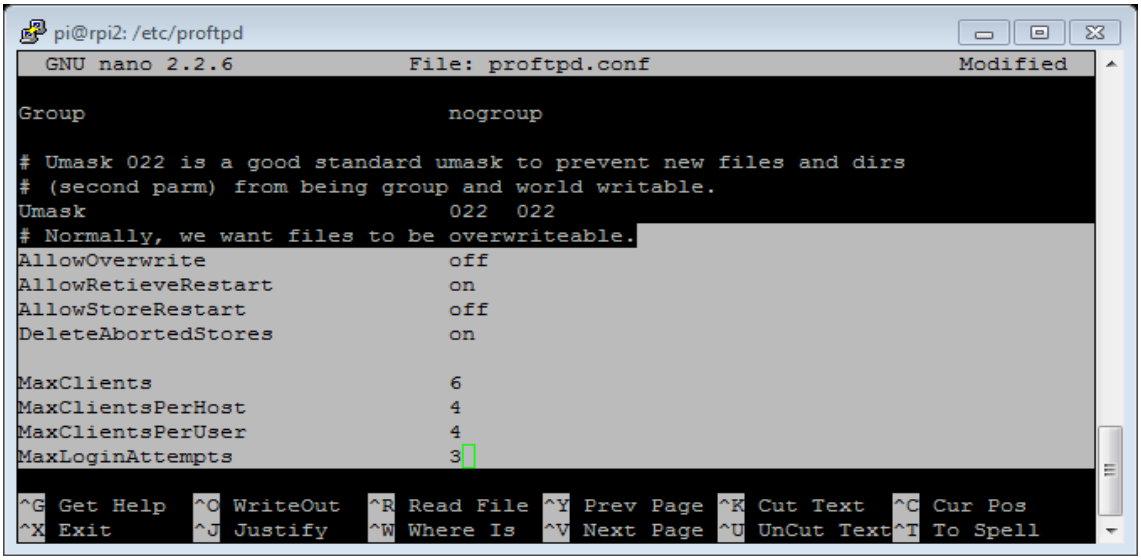


Fig. 59: habilitación y anulación de tareas.

Para culminar con la configuración en este fichero, se debe agregar las líneas que se encuentran señaladas en la imagen:

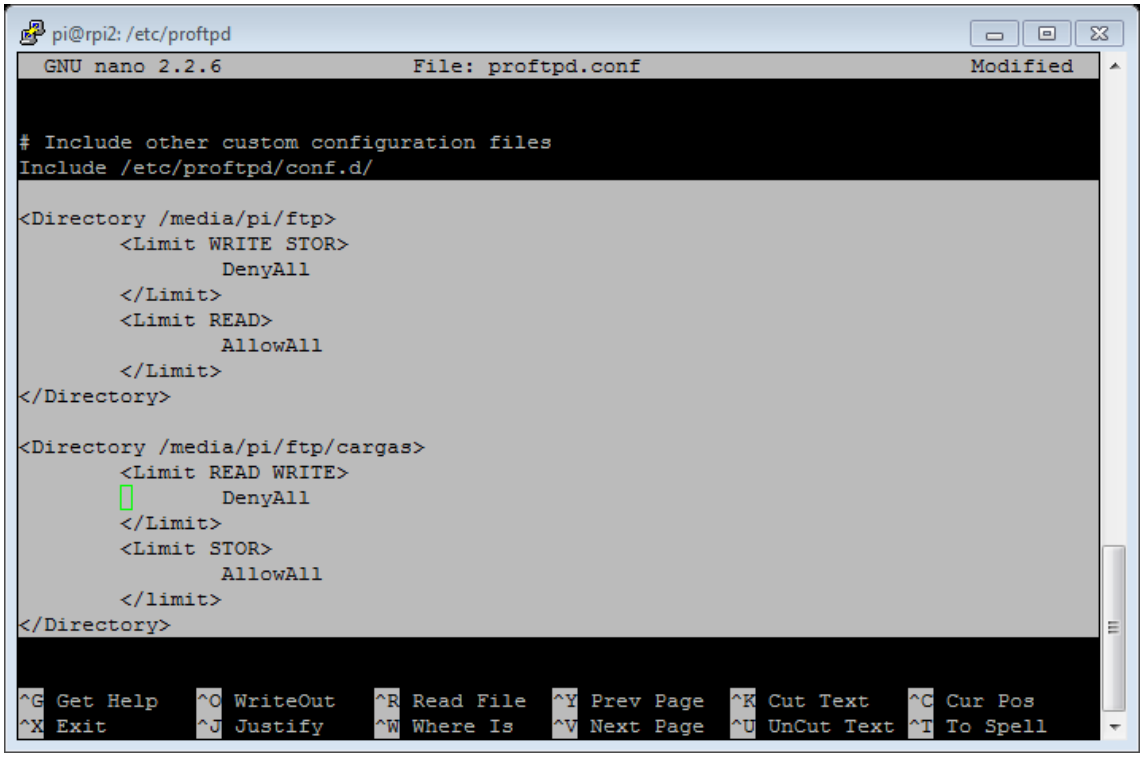


Fig. 60: Definición de permisos para las carpetas.

Estas líneas definen los permisos que tendrán las carpetas de descarga, los usuarios podrán descargar, pero no podrán modificar, así mismo borrarán o enviarán ficheros.

Una vez en la terminal se ejecuta la siguiente línea de comando:

```
sudo /etc/init.d/proftpd restart
```

Esta línea se la ejecuta con la única finalidad de poner en marcha el servidor **ftp** la imagen será la siguiente.

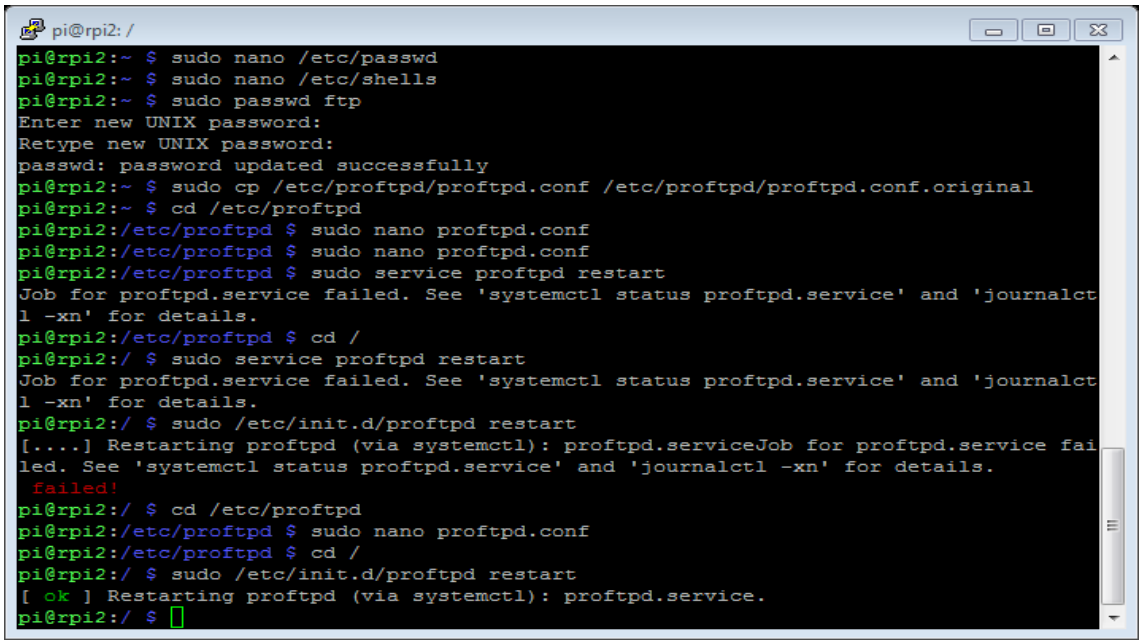


Fig. 61: Puesta en marcha del servidor FTP.

Es posible verificar que las configuraciones hechas en los ficheros, dieron los resultados esperados en la configuración del servidor FTP. Simplemente al ingresar al navegador y ejecutar la siguiente instrucción *ftp://192.168.2.203*, luego pedirá ingresar un usuario y una contraseña, que fueron creados con anterioridad, lo que producirá la visualización de los archivos antes mencionados.

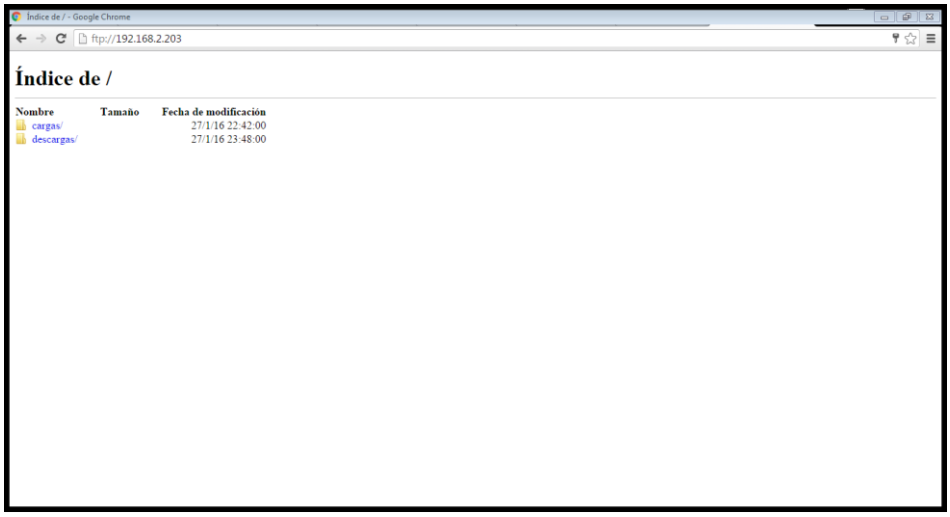


Fig. 62: Ingreso al servidor FTP.

3.6 MySQL

En esta sección se describe como montar las replicaciones de la base de datos MySQL, además de como configurarlas. Esto ayuda a crear una copia exacta desde un servidor maestro a otro servidor esclavo, logrando así, que todas las actualizaciones o modificaciones que se realicen en el servidor principal se reflejen en tiempo real en los nodos alternos, obteniendo como resultado que todas las bases de datos que se tengan instaladas en el clúster trabajen como una sola, sin pérdida de información, ya sea esta de los sistemas de información, o de la gestión propia de la base de datos. Estas replicaciones de base de datos, no son una copia de seguridad frente a cambios de software, pero la replicación puede ayudar a proteger contra fallos de hardware. Este tema contendrá lo siguiente:

- 3.6.1 Instalar MySQL
- 3.6.2 Mover carpeta MySQL en la Unidad externa

3.6.1 Instalar MySQL

Lo primero que se realiza es la instalación de MySQL en todos los nodos del clúster, para ello se ejecuta la siguiente línea.

```
sudo apt-get install mysql-server
```

Aquí se instalan todos los paquetes de MySQL como el: MySQL client, MySQL-Server, entre otros paquetes. Una vez acabado de instalar los paquetes, muestra la siguiente figura.

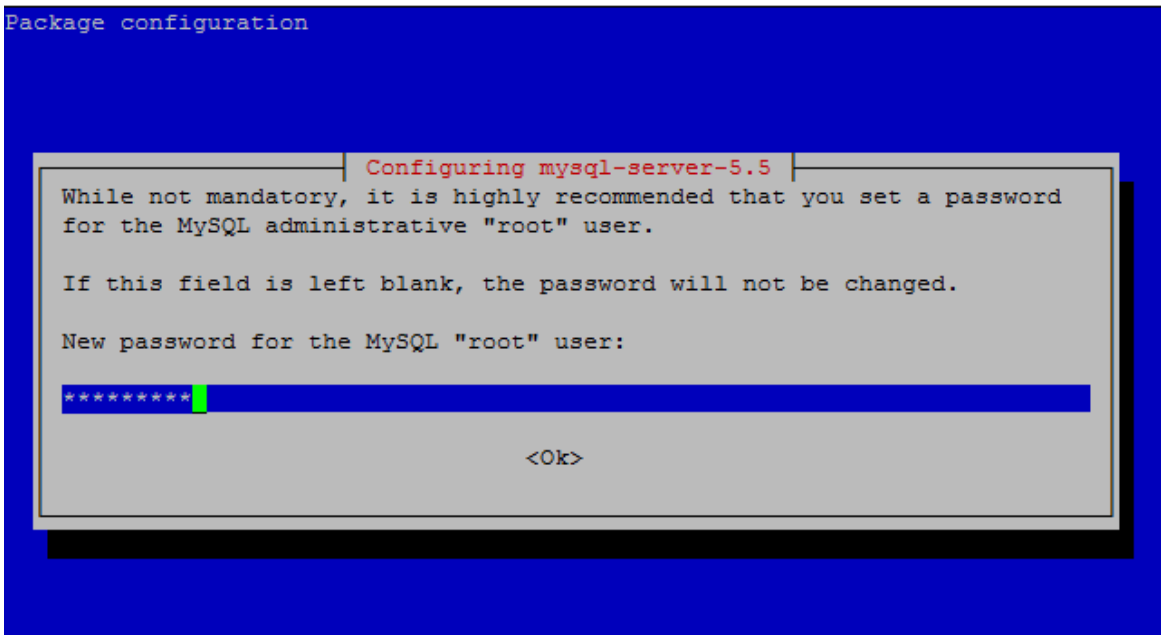


Fig. 63: Ingreso de clave para la configuracion de MySQL-Server.

Esta imagen pide que ingrese una contraseña para el usuario root de MySQL, la clave puede ser elegida por el usuario, esta clave debe ser recordada, ya que más adelante en la configuración de los espejos de la base de datos se necesita de ella. Para este caso de estudio será (raspberry). A continuación pedirá confirmar la contraseña, mostrando una imagen similar, en donde ingresará nuevamente la misma clave (raspberry), para seguir con la instalación.

Por otra parte se debe modificar el fichero */etc/mysql/my.cnf* , con la finalidad de poder acceder remotamente a la base de datos, para ello se edita el fichero.

```
sudo nano /etc/mysql/my.cnf
```

Se busca la línea *bind-address* que por defecto aparece de la siguiente manera *#bind-address =127.0.0.1*, se le quita el comentario, si lo está, y se ingresa la ip del nodo que se va a configurar, todo lo anteriormente indicado quedará como se muestra a continuación.

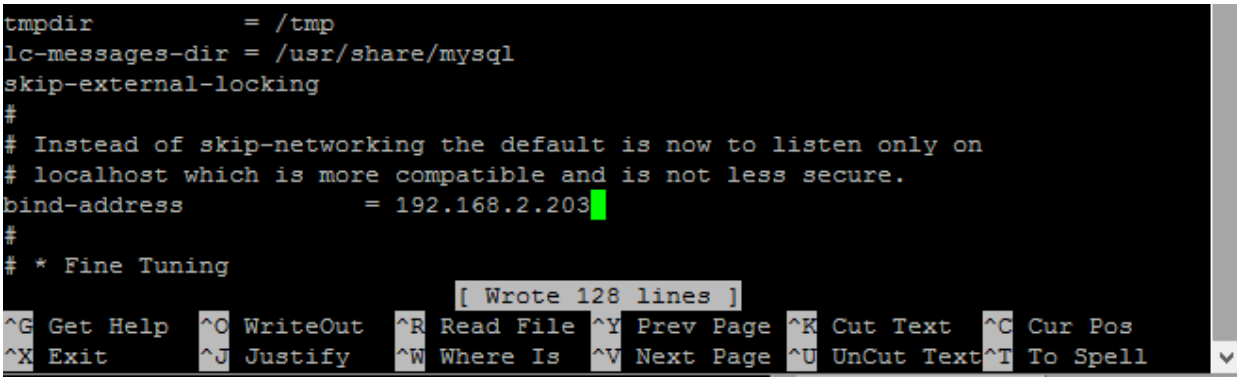


Fig. 64: Configuración de Ip para acceder remotamente a MySQL.

Se procede a guardar los cambios y permaneciendo en el directorio de MySQL, se reinicia el servicio con la siguiente instrucción.

```
sudo /etc/init.d/mysql restart
```

Si no existió ningún problema, se deberá poder acceder a MySQL ejecutando.

```
mysql –u root –p
```

Una vez ejecutado esto, aparecerá una imagen como la que se muestra a continuación indicando que se debe ingresar la contraseña que se configuró pasos atrás en la instalación de MySQL y que es: (raspberry).

Fig. 65: autentificación de acceso a MySQL.

Una vez posesionados en el directorio de MySQL , se debe proceder a dar todos los privilegios para acceder a la base de datos remotamente, para ello se ejecutara las siguientes líneas:

```
GRANT ALL PRIVILEGES on *.* TO 'root'@'%' IDENTIFIED BY
'password' WITH GRANT OPTION;

FLUSH PRIVILEGES;
```

El comando *FLUSH PRIVILEGES*; se lo utiliza para limpiar el cache que se hubieren quedado de algunas otras instalaciones de MySQL o simplemente residuos que puedan generar algún problema, ya que MySQL mantiene una copia de memoria de las tablas de permisos de usuarios.

3.6.2 Mover carpeta MySQL en la Unidad externa

Primeramente para poder mover la carpeta, se debería haber creado con anticipación una carpeta en la USB, a la cual se moverá la carpeta de MySQL, para este caso la carpeta ya fue creada con anterioridad, por lo tanto queda ejecutar la siguiente línea de comando tal y como se muestra en la imagen.

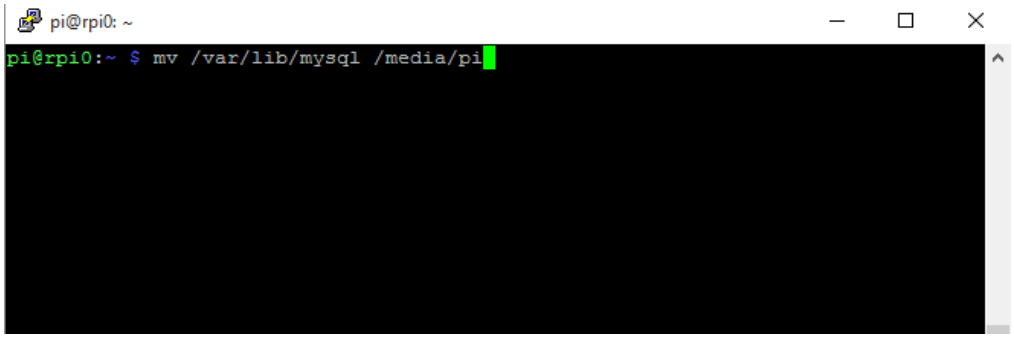


Fig. 66: Mover BD .

Se modifica el fichero *my.cnf*, para que la información que se desea: actualizar, modificar, combiar, o elimiar, se la tome desde la base de datos que ahora se encuentra en la unidad externa. Para ello se ejecuta lo siguiente:

```
sudo nano /etc/mysql/my.cnf
```

Luego se procederá a modificar la línea en donde se encuentra el cursor ubicado, dejándolo tal y como lo indica la figura.

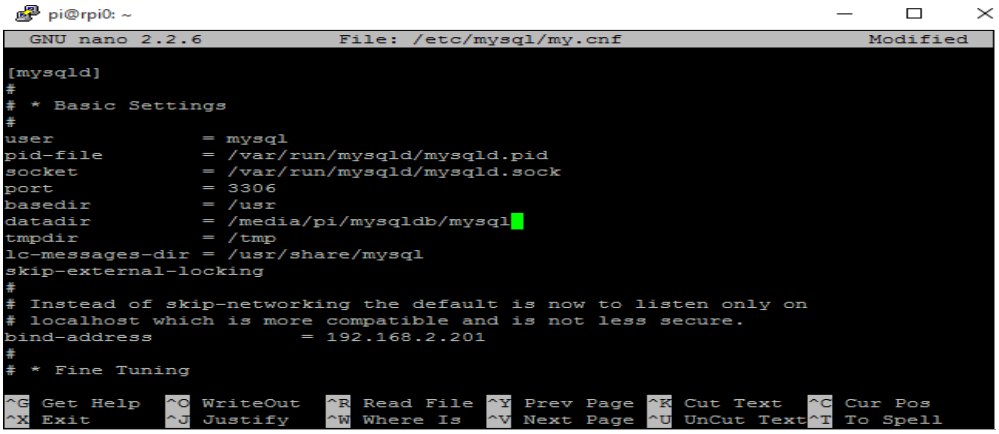


Fig. 67: ingreso de la nueva ubicación de la carpeta de MySQL.

A continuación se debe reiniciar MySQl para que los cambios tengan efecto.

```
sudo /etc/init.d/mysql restart
```

En este momento ya se puede crear una carpeta en la unidad externa para almacenar las bases de datos, para el ejemplo mysqldb.

```
mkdir  mysqldb.
```

Y se procede a dar todos los permisos a la carpeta.

```
chmod 777 mysqldb.
```

3.7 Alta Disponibilidad

Uno de los objetivos específicos de este tema investigativo, es lograr generar pruebas de rendimiento y alta disponibilidad en el clúster, creado con los dispositivos RaspberryPi2. Para lograr esto se necesita de la instalación de programas como el Pacemaker y Cronosync, el Pacemaker administra los nodos y los recursos mientras que el Cronosync, los comunica a cada uno de ellos, para verificar su estado, estos dos programas van de la mano en el proceso de implementación de alta disponibilidad. Los subtemas a tratar en la implementación de la alta disponibilidad serán:

- 3.7.1 Instalación del Pacemaker y Cronosync
- 3.7.2 Configuración de la IP Virtual
- 3.7.3 Levantamiento de servicios Pacemaker y Cronosync
- 3.7.4 Configurar Pacemaker

3.7.1 Instalación del Pacemaker y Cronosync

Los dos programas se instalaran en cada uno de los nodos que se va hacer la prueba de alta disponibilidad, para el caso de estudio se utilizará el nodo rpi1, y rpi2 que son los nodos esclavos en el sistema de red del cluster, y que tienen las ips: 192.168.2.202, 192.168.2.203, respectivamente. Se abre una terminal y se ejecuta lo siguiente.

```
sudo apt-get install pacemaker corosync
```

Se procede a ejecutar la misma línea de código en el otro raspberry rpi2. Luego se debe crear la clave de autentificación en rpi1 del corosync. Ejecutando lo siguiente:

```
sudo corosync-keygen
sudo scp /etc/corosync/authkey 192.168.2.203:/etc/corosync/authkey
```

Con el comando scp se procede a copiar la clave de autentificación de *corosync*, creada en el rpi1 hacia el rpi2. Ahora se debe situar sobre una terminal del rpi2 y se procede a darle todos los permisos a la clave copiada con la siguiente sentencia.

```
sudo chmod 777 /etc/corosync/authkey
```

Una vez ejecutado esto se procede a configurar la red que se va a utilizar, para contralar el estado de los nodos. Para ello se debe editar el fichero */etc/corosync/corosync.conf* tanto para rpi1, y rpi2. Y quedará como se muestra a continuación:

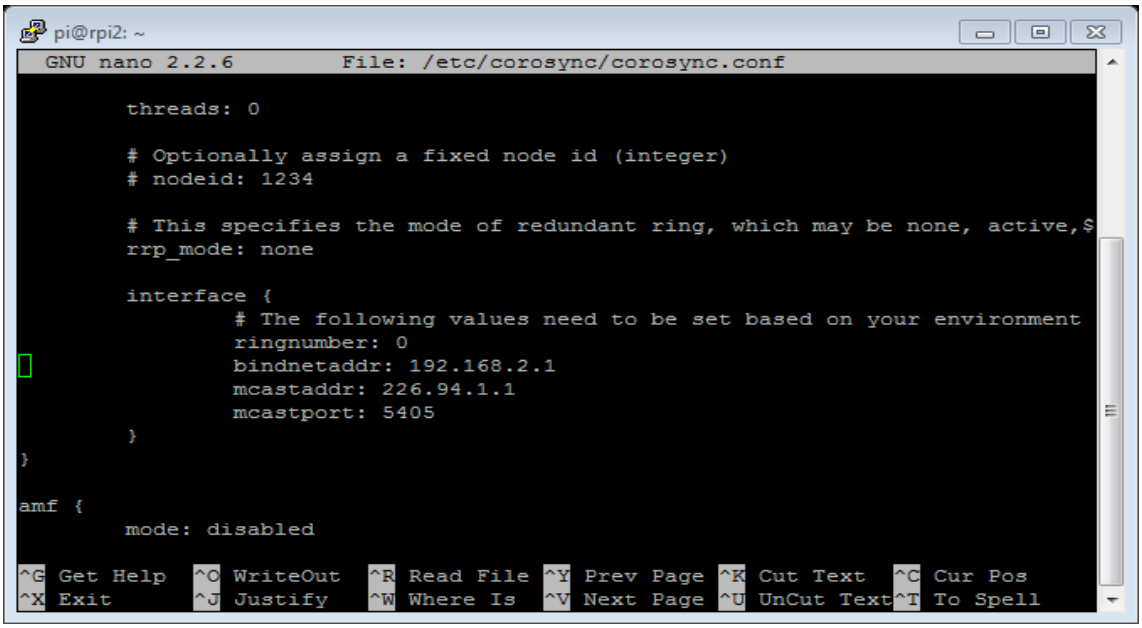


Fig.68: ingreso de la red que verificará el estado de los nodos.

Para que el corosync se inicie automáticamente en el momento en que se reinicia el dispositivo, se debe editar el fichero */etc/default/corosync* y se cambia a YES el START, tal y como lo muestra la figura.

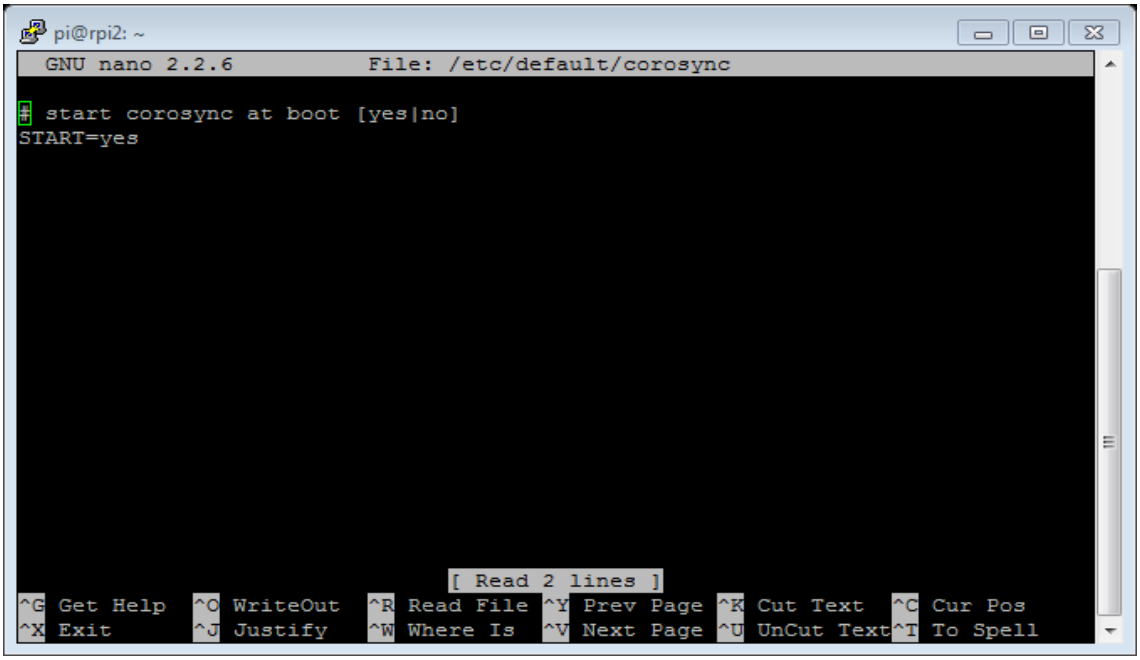


Fig. 69: iniciación del corosync automáticamente.

Para que los cambios tengan efecto, se ejecuta la siguiente línea de comando en cada uno de los nodos.

```
sudo service corosync restart
```

Ahora para comprobar el estado del cluster, se debe ejecutar: *sudo crm_mon,* y mostrará una imagen como la que se muestra a continuación, indicando el estado de los nodos, si están en online u OFFLINE y si el FAILOVER-ADDR está asociado o no al nodo caído. Esto se lo debe realizar con la finalidad de hacer las pruebas de alta disponibilidad en el momento de que un nodo falle.

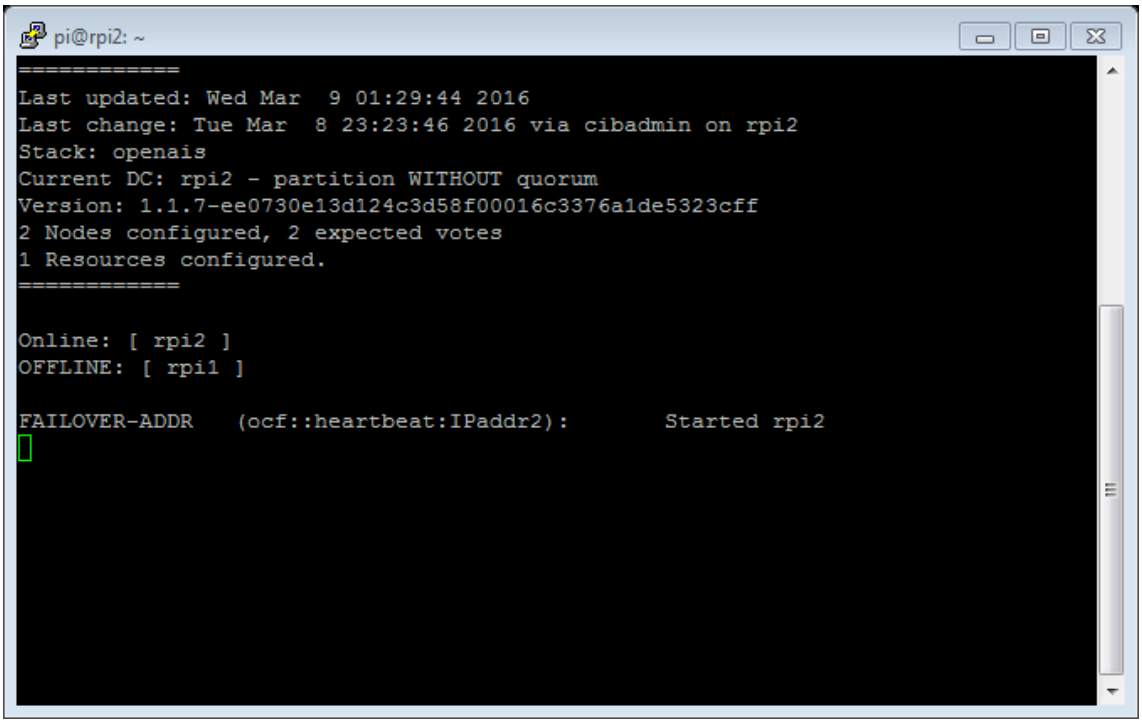


Fig. 70: Prueba de alta disponibilidad.

3.7.2 Configuración de la IP Virtual

Una IP virtual es una buena forma de tener acceso de comunicación sin la necesidad de enlazar la dirección con alguna interfaz física, estas IP son utilizadas en diversas formas de configuración de los clústers, por lo que se convierte en un circuito de entrada y salida de información.

La IP virtual en el caso de estudio, servirá para levantar los servicios tanto MySql como FTP, que serán configurados más adelante con el fin de construir la alta disponibilidad y rendimiento del clúster. La IP Virtual ofrecerá la posibilidad de levantar cualquiera de los servicios en cualquiera de los dos nodos, en caso de alguna

caída o problema con alguno de ellos, esta será el nexo de comunicación auxiliar entre cada uno de los ellos, la ip será configurada junto con el Pacemaker y Cronosync.

Para hacer la sincronización de la Ip Virtual con el Pacemaker solo se debe ejecutar lo siguiente:

Primero se deshabilita el mecanismo stonith desde el rpi1, para así evitar conflictos en el nodo o nodos que estén dando problemas, para ellos se ejecuta lo siguiente.

```
sudo crm configure property stonith-enabled=false
```

Una vez ejecutado el comando anterior, se pasa a la configuración de la Ip con lo siguiente instrucción.

```
sudo crm configure primitive FAILOVER-ADDR ofc:heartbeat:Ipaddr2 params ip=”192.168.2.205” nic=”eth0” op monitor interval=”10s” meta is-managed=”true”
```

Para hacer las pruebas de verificación, simplemente se debe hacer un ping a la ip virtual que se ha configurado.

```
sudo ping 192.168.2.205
```

3.7.3 Levantamiento de servicios Pacemaker y Cronosync

Para que exista una sincronización entre los servicios de la alta disponibilidad y los servicios que se requiere levantar, es necesario hacer la configuración necesaria para ello. Pero antes de empezar, se necesita estar seguro que exista una comunicación entre nodos interconectados por el Cronosync. Estas pruebas se las puede hacer revisando los pasos *3.6.1 Instalación del Pacemaker y Cronosync*, de este documento y verificar si las configuraciones se han hecho de la manera correcta, y están de acuerdo a lo que se requiere.

3.7.4 Configurar Pacemaker

Lo que realmente va hacer el Pacemaker dentro de la configuración del clúster, es verificar la caída de un nodo, como su nombre lo indica es un marca pasos que está verificando la latencia de los nodos, y dar una señal de alerta si no existen los pulsos de verificación en alguno de ellos. Para configurarlo se va a utilizar la sentencia *crm*. Si por ejemplo se necesita verificar el estado de los nodos, se ejecuta la siguiente línea:

```
sudo crm node show
```

Y mostrará una imagen como la siguiente, en donde indica el normal funcionamiento, caso contrario mostrará un fallo.

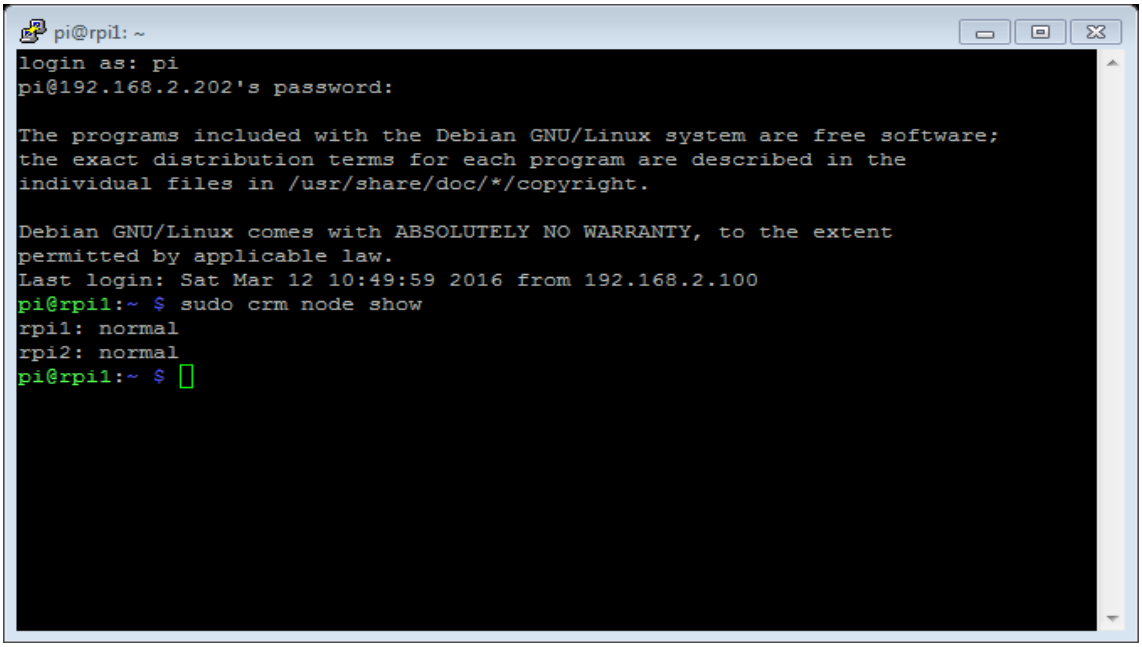


Fig. 71: Prueba del normal funcionamiento de los Raspberrys.

Algo importante que hay que verificar es el funcionamiento de la clase y del proveedor del servicio por ello se ejecuta lo siguiente:

```
sudo crm ra clases
```

Mostrará la siguiente:

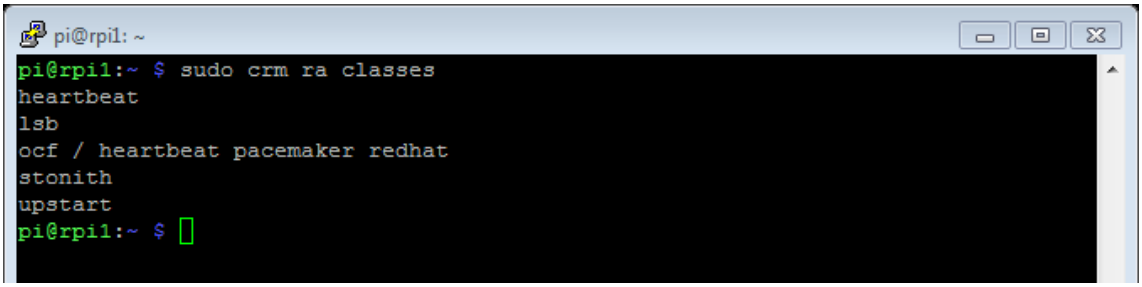


Fig. 72: Muestra las clases y proveedores en ejecución.

Para configurar el servicio que se necesita que Pacemaker levante, es necesario previamente saber el listado de parámetros a configurar, para ello se ejecuta la siguiente instrucción:

```
sudo crm ra meta mysql
```

Mostrará una imagen como la siguiente, indicado todos los parámetros para configurar en cada uno de los servicios, para este caso de estudio se lo realizó con mysql.

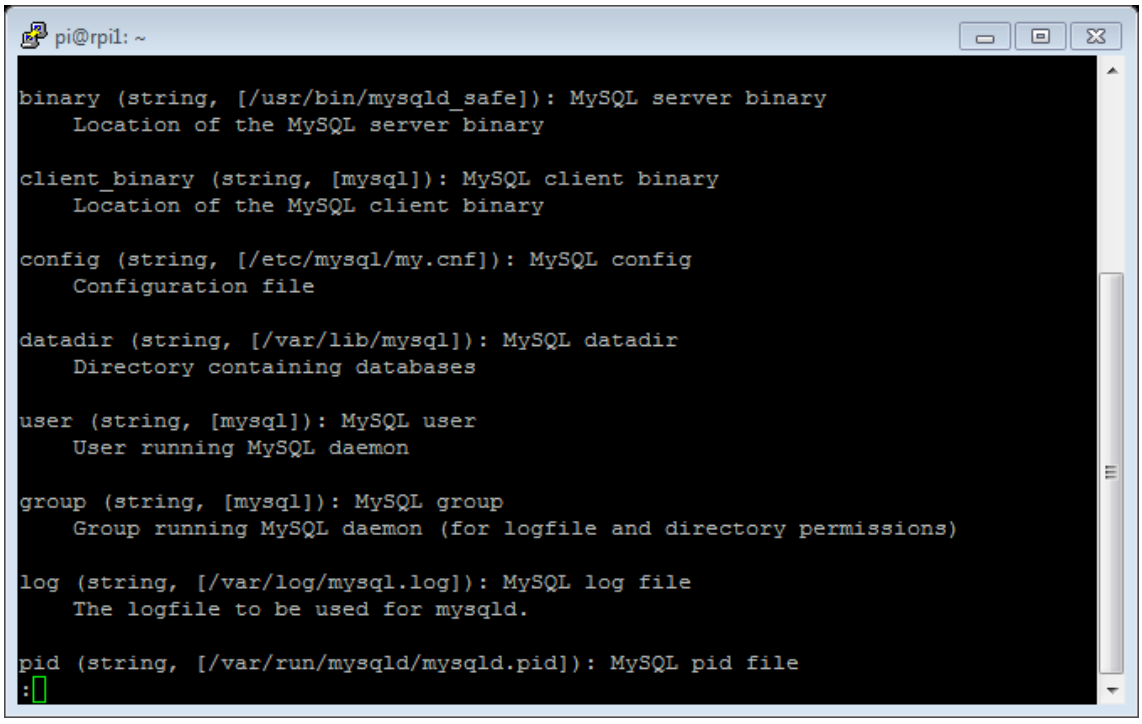


Fig. 73: Muestra parámetros de configuración MySql.

Se configura el servicio ejecutando la siguiente línea de comando con todos los parámetros necesarios tal y como se muestra:

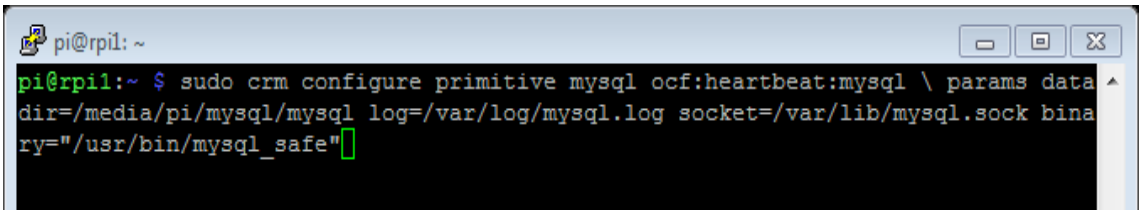


Fig. 74: Configuración de parámetros de MySql necesarios para levantar el servicio en Pacemaker.

Se procede a ejecutarla y de haber algún error en la configuración se debe editar el fichero de configuración con la siguiente línea de comando.

```
sudo crm configure edit
```

Indicará todo la configuración, que puede ser modificada. También es necesario configurar los tiempos de monitoreo para cada recurso, como lo muestra la figura.

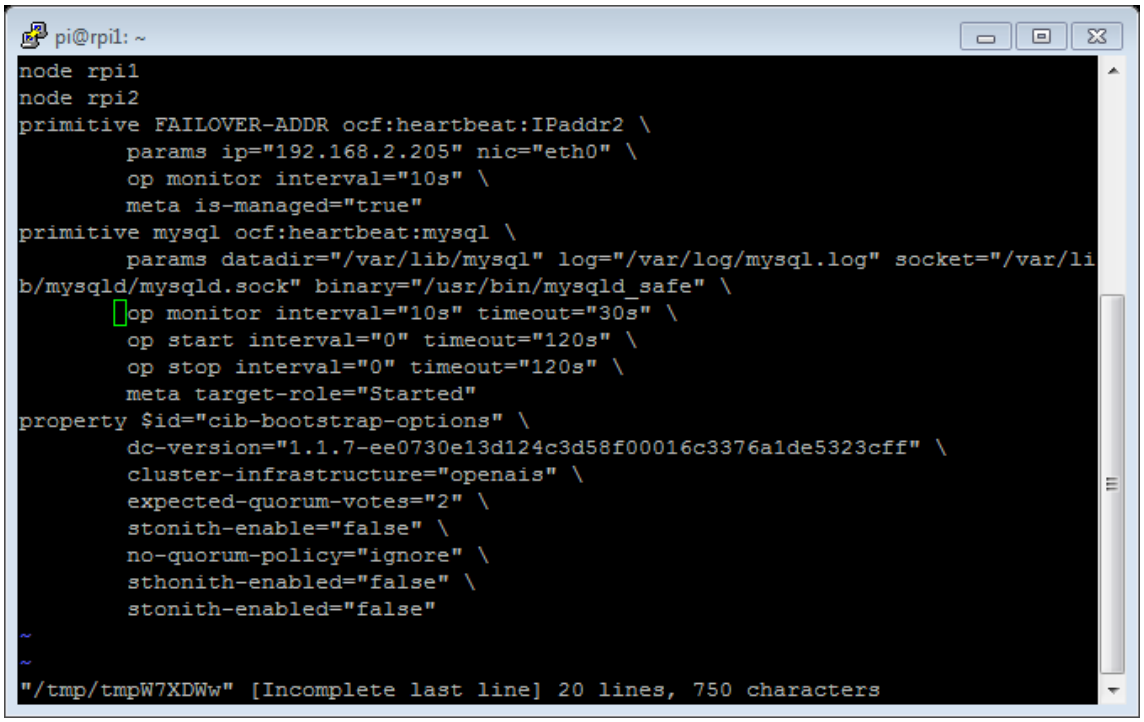


Fig. 75: Archivo de configuración de levantamiento de servicio MySql.

Faltaría únicamente verificar el funcionamiento de la configuración, de no haber existido errores mostrará una imagen como esta:

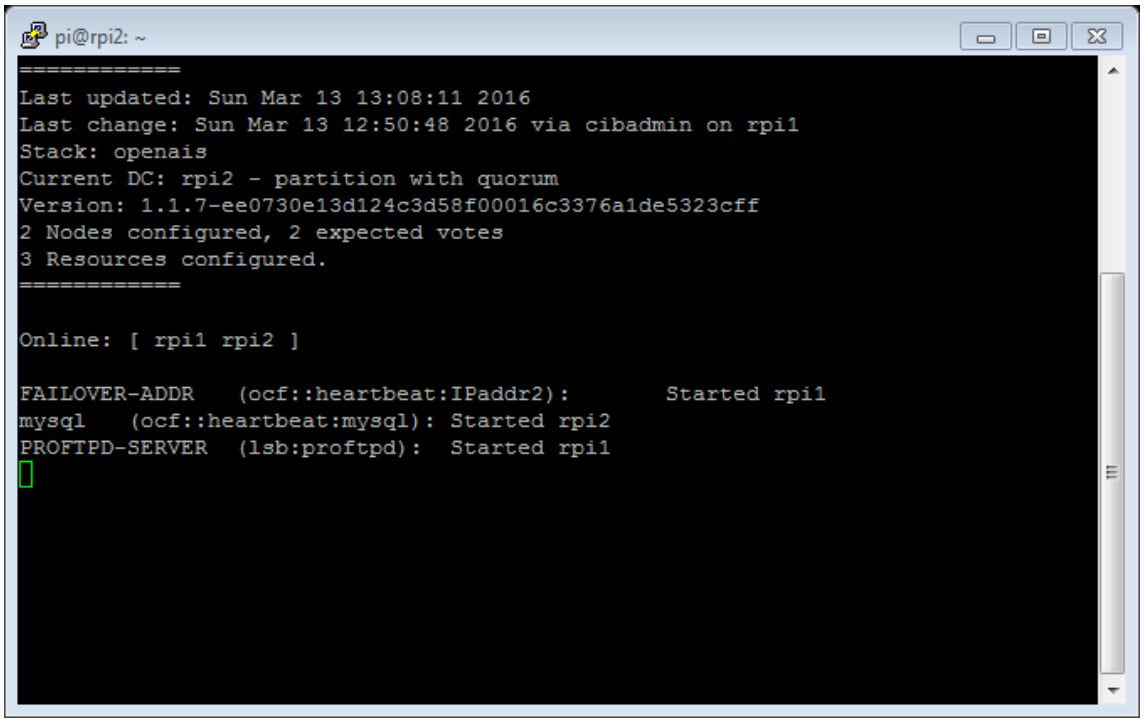


Fig. 76: Verificación y Funcionalidad de Servicios en Pacemarket.

Se deben hacer las pruebas necesarias, como hacer caer cualquiera de los nodos para así comprobar la funcionalidad de la alta disponibilidad.

Realizando estas pruebas se comprobó que existe problemas de configuración en Pacemaker para que levante el servicio de MySql, ya que al hacer las pruebas tales como: hacer caer un nodo y hacer consultas en la base de datos, muestra un error, no ejecuta la sentencia MySql e intenta acceder localmente a la instancia de MySql, creando conflictos en medios compartidos. Pero al verificar el estado de la base de datos en el nodo, el Pacemaker muestra como si estuviera activa, tal y como lo muestra la imagen Fig.76.

Este error no se logró depurar y queda abierto para nuevos estudios, se continua con el desarrollo del proyecto, por lo que al probar estos servicios independientemente se llegó a mostrar la alta disponibilidad y su normal funcionamiento, la conclusión es una incompatibilidad entre MySQL y Pacemaker o la falta de una librería adicional para lograr su sincronización. Esto se pude deber a un bug en la arquitectura del depósito Raspberry Pi2 al emplear Pacemaker, se debería realizar un estudio más extenso, para encontrar una solución y montar un sistema completo de alta disponibilidad.

Continuando con el desarrollo, para levantar otro servicio se debe hacer lo mismo que se ha hecho desde el paso 3.7.1, solamente es necesario verificar bien los parámetros del servicio y configurarlos tal y como se lo hizo con MySql.

Se inserta la primitiva del PROFTPD para el servicio FTP, tal y como lo indica la imagen, y puntualmente el cursor.

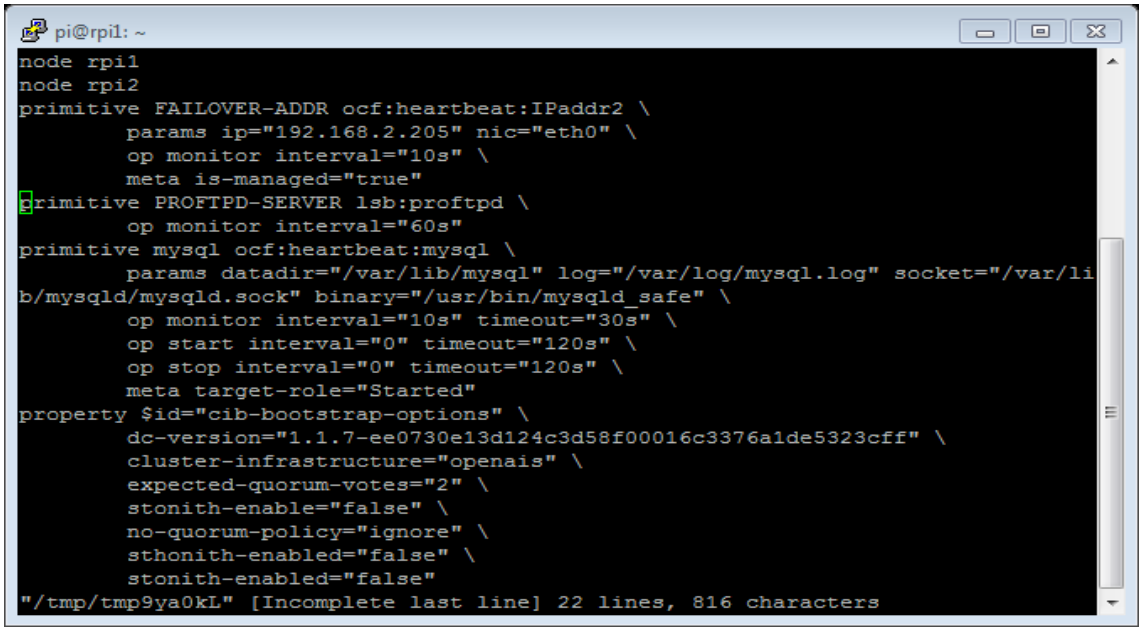


Fig. 77: Archivo de configuración de levantamiento de servicio FTP.

Se procede hacer las pruebas de verificación de levantamiento de servicio FTP, las mismas pruebas que se hizo con MySQL, estas pruebas se las realizó manualmente y con la ayuda del software Filezilla, Las pruebas realizadas fueron copia y pega de archivos, conectar y desconectar los nodos, y su funcionamiento fue satisfactorio. La imagen que corrobora que no existe ningún error y que la alta disponibilidad y rendimiento funcionan con FTP, es la imagen siguiente.

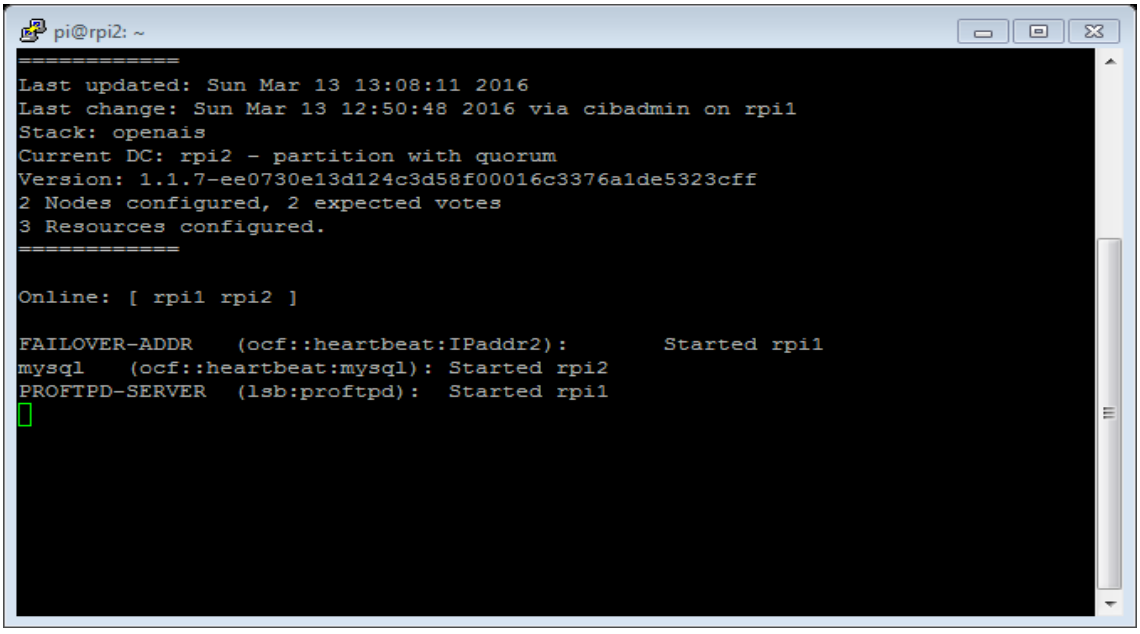


Fig. 78: Verificación y Funcionalidad de Servicios en Pacemarket.

Con las pruebas realizadas, y con los inconvenientes puestos a consideración, se ha llegado a cumplir con el objetivo, y con ello la culminación de este trabajo de titulación.

Conclusiones

El objetivo fundamental de esta tesis fue implementar un sistema cliente-servidor, a un costo más asequible para las empresas pymes, y así aportar con una solución que propicie la reducción de la brecha tecnológica de estas organizaciones frente a las denominadas grandes. El clúster está compuesto por tres dispositivos RaspberryPi2, un switch, cables conectores utp, memorias y requiere de configuraciones en cada uno de los equipos para que estos funcionen como nodos, y a su vez trabajen como uno solo.

Se ha escogido RasperryPi2 como un medio de desarrollo ya que, es un dispositivo que ofrece una solución rentable en la implementación del clúster, a diferencia de un clúster tradicional. Los clústers tradicionales necesitan a más de un presupuesto alto, un ambiente apropiado para su implementación, llamado cuarto frio. Este cuarto frio a más de los equipos necesarios para armar el clúster, necesita de otros equipos adicionales que garanticen su normal funcionamiento. Los clústers diseñados mediante esta nueva tecnología *RaspberryPi2*, por sus características y por ser muy ligeros, no necesitan de estas condiciones, lo que ayuda a optimizar recursos y espacio.

La motivación que conllevó a realizar este tema de tesis, es el estar inmiscuido en el mercado laboral de telecomunicaciones y con empresas que requieren de este servicio. El entusiasmo de crear nuevas opciones de desarrollo de sistemas cliente-servidor con nuevas tecnologías y dispositivos que garanticen prestaciones más asequibles que los tradicionales.

Los dispositivos tecnológicos, entre ellos RaspberryPi2, son un factor esencial, que fomentan el desarrollo educativo por su gran impacto y rápida expansión a nivel mundial, pues esta placa fue creada para el desarrollo de proyectos estudiantiles. Además, este tipo de dispositivos, generan un crecimiento productivo mediante un incremento en la eficiencia, calidad y diferenciación de los modelos de clústers construidos con esta tecnología.

La implementación de estas tecnologías tienen un rango de incertidumbre, misma que se llegó a despejar con la implementación del clúster, que cuenta con 3 nodos, 1 maestro y dos esclavos, en el maestro está montada una unidad USB, que es en donde se muestran todos los servicios, además se ha instalado servicios de Base de Datos y Servidores en cada uno de los nodos esclavos, para realizar las pruebas de rendimiento y alta disponibilidad

Este trabajo deja la puerta abierta a que nuevas personas lo retomen para realizar otros proyectos, aplicando nuevas y mejores tecnologías, por lo que quedaron temas por implementar como: computación paralela, espejos de servicios, respaldo de los nodos, entre otros, que pueden ser implementadas con nuevos y mejores equipamientos, puesto que en la actualidad ya se encuentra en el mercado el RasperryPi3 que tiene mejores características y prestaciones de desarrollo que el RaspberryPi2 con el que se desarrolló este trabajo.

En la mayoría de los trabajos de investigación, todo lo que se hace por primera vez, no siempre llega a tener los resultados esperados, como ocurrió en este caso, la falta de tiempo y desconocimiento de algunos temas, hizo que se vea solo la punta del iceberg, en lo relacionado a la utilización de RaspberriPi2 y de sus funcionalidades en el desarrollo del clúster, dejando así temas inconclusos como la implementación de servicios como: MySQL, Apache, entre otros, y que se podrán seguir desarrollando y mejorando con nuevas y mejores implementaciones.

Lo más importante es haber llegado al objetivo propuestol que era: “Implementar un sistema cliente-servidor mediante tecnología *RaspberryPI2* que garantice los mismos beneficios de un servidor tradicional a un costo asequible a empresas e instituciones de tipo pymes.”. Pero cabe recalcar que este objetivo se ha cumplido con la implementación y prueba de ciertos servicios como el FTP, y que en servicios como el MySQL no se lo pudo lograr, por no existir una sincronización en los procesos de MySql y Pacemarket, que individualmente funcionan, pero en conjunto faltó la configuración necesaria poder levantar este servicio y lograr hacer las pruebas de alta disponibilidad y rendimiento.

Glosario

SIG: dentro de Cliente_sevidor significa Sistemas de Información Gerencia

NFS: (Network File Services) dentro de cliente servidor

Mainframe: Mainframe es un gran ordenador capaz de realizar el procesamiento de datos complejos.

Intranet: Una red configurada internamente exclusiva para que tengan acceso los empleados de la organización.

HTTP: (Hyper Text Transfer Protocol).

SQL: (*Structured Query Language*)

OLTP: (**O**n**L**ine **T**ransaction **P**rocessing), es un procesamiento que facilita y administra aplicaciones transaccionales.

TIC: (Tecnologías de Información y la Comunicación)

MTTF: (Mean Time To Failure): Promedio de tiempo entre fallos.

MTTR: (Mean Time To Recover): Tiempo medio de recuperación.

Bibliografía

Blaise Barney, L. L. (17 de 11 de 2015). *Introduction to Parallel Computing*. Obtenido de https://computing.llnl.gov/tutorials/parallel_comp/#MemoryArch

Brash, D. (2010). Extensions to the ARMv7-A Architecture. ARM.

Breschi, S. M. (2006). *Clusters, Networks, and Innovation* . Oxford University Press, UK.

Celkos, J. (2011). *SQL for Smarties.* EE.UU: Elsevier.

Chan, P. (2015). *Raspberry Pi User Guide.* Ingaterra: element14.

Costas Santos, J. (2014). *Seguridad y alta disponibilidad.* RA-MA Editorial.

Crespo, C. G. (2014). Diseño y evaluación de un clúster HPC: Aplicaciones. *UPC*, 80.

Dang, B. G. (2014). *Practical Reverse Engineering : Using X86, X64, ARM, Windows Kernel, and Reversing Tools* . John Wiley & Sons, Incorporated.

Dennis, A. K. (2013). *Raspberry Pi Super Cluster.* Packt Publishing Ltd.

Dennis, A. K. (2013). *Raspberry Pi Super Cluster .* Packt Publishing Ltd.

Joubert, G. R. (2002). *Parallel Computing : Advances and Current Issues : Proceedings of the International Conference, PARCO 2001 .* Imperial College Press.

Kiepert, J. (2013). Creating a Raspberry Pi-Based Beowulf Cluster. *RpiCluster*, 30.

Muñoz Escoí, F. D. (2013). *Concurrencia y sistemas distribuidos.* Editorial de la Universidad Politécnica de Valencia.

n.d. (2013). Raspberry Pi. *sheffieldlearningcommunity*, 1045.

Pi, R. (18 de 12 de 2013). *Raspberry Pi, historia Informática.* Recuperado el 28 de 10 de 2015, de http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/

Pi, R. (18 de 12 de 2013). *Historia de la Informatica.* Recuperado el 28 de 10 de 2015, de http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/

Raspberrypi. (2010). *MAKE RESOURCES*. (Raspberrypi) Recuperado el 13 de 10 de 2015, de https://www.raspberrypi.org/resources/make/

Raspberrypi. (2010). *PICADEMY*. (Raspberrypi) Recuperado el 13 de 10 de 2015, de https://www.raspberrypi.org/picademy/

Raspberrypi. (2010). *TEACH RESOURCES*. (Raspberrypi) Recuperado el 13 de 10 de 2015, de https://www.raspberrypi.org/resources/teach/

raspberrypi. (2010). *WHAT IS A RASPBERRY PI?* (raspberrypi) Recuperado el 14 de 10 de 2015, de https://www.raspberrypi.org/help/what-is-a-raspberry-pi/

Robinson, A. C. (2013). *Raspberry Pi Projects .* John Wiley & Sons, Incorporated.

Rraspberrypi. (2010). *LEARN RESOURCES*. (Rraspberrypi) Recuperado el 13 de 10 de 2015, de https://www.raspberrypi.org/resources/learn/

Yadav, S. C. (2009). *Introduction to Client Server Computing .* New Age International.

Doctora Jenny Ríos Coello, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay,

C E R T I F I C A:

Que, el Consejo de Facultad en sesión del 05 de noviembre de 2015, conoció la petición del (los) estudiante(s) **Rubén Omar Riofrío Ocaña y Fausto Vinicio Sarango Molina** con código(s) **34526 y 34314 respectivamente**, registrado(s) en la Unidad de Titulación Especial, quien(es) denuncia(n) su trabajo de titulación denominado: **"CLUSTER DE SERVIDORES MEDIANTE DISPOSITIVOS RASPBERRYPi2"**en la modalidad: Proyecto de investigación y presentado como requisito previo a la obtención del título de Ingenieros de Sistemas .-El Consejo de Facultad acoge el informe de la Junta Académica y aprueba la denuncia. Designa como Director(a) a Ing. Marcos Orellana Cordero y como miembro del Tribunal Examinador a Ing. Francisco Salgado Arteaga. De conformidad con el cronograma de la Unidad de Titulación el (los) peticionario(s) debe presentar su trabajo de titulación hasta el 11 de marzo de 2016.

Cuenca, 06 de noviembre de 2015

Dra. Jenny Ríos Coello
Secretaria de la Facultad de
Ciencias de la Administración

C O N V O C A T O R I A

Por disposición de la Junta Académica de Ingeniería de Sistemas y Telemática, se convoca a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación: "Clúster de servidores mediante dispositivos RaspberryPi2" presentado por los estudiantes **Fausto Vinicio Sarango Molina** con código 34314 y **Rubén Omar Riofrío Ocaña** con código 34526 previa a la obtención del grado de Ingeniero de Sistemas, para el día **MIERCOLES 28 DE OCTUBRE DE 2015 A LAS 07h30.**

Cuenca, 27 de abril de 2015

Dra. Jenny Ríos Coello
Secretaria de la Facultad

Ing. Marcos Orellana Cordero

Ing. Francisco Salgado Arteaga

Oficio Nro. 153-2015-DIST-UDA

Cuenca, 28 de Octubre de 2015

Señor Ingeniero
Xavier Ortega Vázquez
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMNISTRACIÓN
Presente.-

De mis consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, eunida el día 28 de octubre del 2015, revisó la documentación del proyecto de tesis denominado "Clúster de servidores mediante dispositivos RaspberryPi2", presentado por los estudiantes Rubén Omar Riofrío Ocaña y Fausto Vinicio Sarango Molina, estudiantes de la Escuela de Ingeniería de Sistemas y Telemática, previo a la obtención del título de Ingeniero de Sistemas.

La Junta considera que la documentación cumple con las normas legales y reglamentarias de la Universidad y de la Facultad de Ciencias de la Administración y avala la aprobación por parte del tribunal designado, así por su digno intermedio, el conocimiento y aprobación por parte del Consejo de Facultad.

Atentamente,

Ing. Marcos Orellana Cordero
Director Escuela de Ingeniería de Sistemas y Telemática
Universidad del Azuay

Catalina Astudillo

Oficio Nro. 152-2015-DIST-UDA

Cuenca, 28 de Octubre de 2015

Señor Ingeniero
Xavier Ortega Vázquez
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMNISTRACIÓN
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 28 de octubre del 2015, recibió el proyecto de tesis titulado "Clúster de servidores mediante dispositivos RaspberryPi2", presentado por los estudiantes Rubén Omar Riofrío Ocaña y Fausto Vinicio Sarango Molina, estudiantes de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por el Ing. Marcos Orellana, previo a la obtención del título de Ingeniero de Sistemas.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomienda como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior a al Ing. Marcos Orellana y como miembro del Tribunal a Francisco Salgado Ph.D.

Atentamente,

Ing. Marcos Orellana Cordero
Director Escuela de Ingeniería de Sistemas y Telemática
Universidad del Azuay

Catalina Astudillo

UNIVERSIDAD DEL AZUAY

ACTA
SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

1.1 **Nombre del estudiante:** Fausto Vinicio Sarango Molina y Rubén Omar Riofrío Ocaña Código 34314 y 34526
1.2 **Director sugerido**: Ing. Marcos Orellana Cordero
1.3 **Codirector (opcional):** _____
1.4 **Tribunal:** Ing. Francisco Salgado Arteaga
1.5 **Título propuesto:** "Clúster de servidores mediante dispositivos RaspberryPi2"
1.6 **Resolución**:

1.6.1 Aceptado sin modificaciones ✓

1.6.2 Aceptado con las siguientes modificaciones:

_____

_____

_____

1.6.3 Responsable de dar seguimiento a las modificaciones: Ing. Marcos Orellana Cordero
1.6.4 No aceptado
- Justificación:

_____

_____

_____

Tribunal

Ing. Marcos Orellana Cordero

Ing. Francisco Salgado Arteaga

Sr. Fausto Sarango Molina

Sr. Rubén Omar Riofrío Ocaña

Dra. Jenny Ríos Coello
Secretario de Facultad

Fecha de sustentación: Miércoles 28 de octubre de 2015 a las 07h30

UNIVERSIDAD DEL AZUAY

RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN

1.1 Nombre del estudiante: Fausto Vinicio Sarango Molina y Rubén Omar Riofrío Ocaña
1.1.1 **Código** 34314 y 34526
1.2 Director sugerido: Ing. Marcos Orellana Cordero
1.3 Codirector (opcional):.
1.4 Título propuesto: "Clúster de servidores mediante dispositivos RaspberryPi2"
1.5 Revisores (tribunal): Ing. Francisco Salgado Arteaga
1.6 Recomendaciones generales de la revisión:

	Cumple totalmente	Cumple parcialmente	No cumple	Observaciones (*)
Línea de investigación				
1. ¿El contenido se enmarca en la línea de investigación seleccionada?	✓			
Título Propuesto				
2. ¿Es informativo?	✓			
3. ¿Es conciso?	✓			
Estado del arte				
4. ¿Identifica claramente el contexto histórico, científico, global y regional del tema del trabajo?	✓			
5. ¿Describe la teoría en la que se enmarca el trabajo	✓			
6. ¿Describe los trabajos relacionados más relevantes?	✓			
7. ¿Utiliza citas bibliográficas?	✓			
Problemática y/o pregunta de investigación				
8. ¿Presenta una descripción precisa y clara?	✓			
9. ¿Tiene relevancia profesional y social?	✓			
Hipótesis (opcional)				
10.¿Se expresa de forma clara?				
11.¿Es factible de verificación?				
Objetivo general				
12.¿Concuerda con el problema formulado?	✓			
13.¿Se encuentra redactado en tiempo verbal infinitivo?	✓			
Objetivos específicos				
14.¿Concuerdan con el objetivo	✓			

general?				
15.¿Son comprobables cualitativa o cuantitativamente?	✓			
Metodología				
16.¿Se encuentran disponibles los datos y materiales mencionados?	✓			
17.¿Las actividades se presentan siguiendo una secuencia lógica?	✓			
18.¿Las actividades permitirán la consecución de los objetivos específicos planteados?	✓			
19.¿Los datos, materiales y actividades mencionadas son adecuados para resolver el problema formulado?	✓			
Resultados esperados				
20.¿Son relevantes para resolver o contribuir con el problema formulado?	✓			
21.¿Concuerdan con los objetivos específicos?	✓			
22.¿Se detalla la forma de presentación de los resultados?	✓			
23.¿Los resultados esperados son consecuencia, en todos los casos, de las actividades mencionadas?	✓			
Supuestos y riesgos				
24.¿Se mencionan los supuestos y riesgos más relevantes?	✓			
25.¿Es conveniente llevar a cabo el trabajo dado los supuestos y riesgos mencionados?	✓			
Presupuesto				
26.¿El presupuesto es razonable?	✓			
27.¿Se consideran los rubros más relevantes?	✓			
Cronograma				
28.¿Los plazos para las actividades son realistas?	✓			
Referencias				
29.¿Se siguen las recomendaciones de normas internacionales para citar?	✓			
Expresión escrita				
30.¿La redacción es clara y fácilmente comprensible?	✓			
31.¿El texto se encuentra libre de faltas ortográficas?	✓			

(*) Breve justificación, explicación o recomendación.

- Opcional cuando cumple totalmente,

- Obligatorio cuando cumple parcialmente y NO cumple.

.....

.....

.....

Ing. Marcos Orellana Cordero

Ing. Francisco Salgado Arteaga

UNIVERSIDAD DEL AZUAY

Cuenca, 28 de Octubre del 2015

Señor Ingeniero

Xavier Ortega Vázquez

DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINNISTRACIÓN

Presente.-

De nuestras consideraciones:

Yo Fausto sarango con código 34314 junto con Rubén Riofrío con código 34526, solicitamos a usted que se nos conceda la aprobación de nuestro tema de tesis "Clúster de Servidores mediante Dispositivos RaspberryPi2" que será implementado en el campo Empresarial en el área de Redes y Telecomunicaciones, utilizando el dispositivo RaspberryPi2 para su desarrollo e implementación, previo a la obtención del título de Ingenieros en Sistemas.

Por la atención que preste a nuestra solicitud le damos nuestros más sinceros agradecimientos.

Atentamente,

Fausto Sarango	Rubén Ríofrio
34314	34526
C.I:1103990428	C.I:1102955521

Edición autorizada de 30.000 ejemplares
Del 708.501 al 738.500
Nº 0717559

UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN

ESCUELA DE INGENIERÍA DE SISTEMAS

Diseño de Tesis

Tema:

Clúster de Servidores mediante Dispositivos RaspberryPi2

Integrantes:

Fausto Sarango

Rubén Riofrio

Director propuesto:

Ing. Marcos Orellana Cordero

Universidad del Azuay

Ingeniería de Sistemas y Telemática

Metodología de la Investigación

Guía para la presentación de la propuesta de proyecto de titulación

1. Datos generales

1.1 Nombre de los estudiantes:

Riofrío Ocaña Rubén Omar.

Sarango Molina Fausto Vinicio

1.1.1 Código:

34526

34314

1.1.2 Contacto: teléfonos: convencional, celular y correo electrónico.

07 2341626

0984171937

rubenriofrio@gmail.com

2688432

0986456565

viniciosarango@hotmail.es

1.2 Director sugerido:

Orellana Cordero Marcos Patricio

Ingeniero en Sistemas.

1.2.1 Contacto: marore@uazuay.edu.ec

1.5 ***Tribunal designado:***

1.6 ***Aprobación:***

1.7 Línea de Investigación de la carrera: *Ingeniería de sistemas y telecomunicaciones.*

1.7.1 ***Código UNESCO***

1203.99 Telemática

1.7.2 Tipo de trabajo:

Proyecto de Investigación.

1.8 Área de estudio: Infraestructura Servidores (Hardware y Software.)

1.9 Título propuesto: Clúster de Servidores mediante Dispositivos *RaspberryPi2*

1.10 Subtítulo: Implementación de un sistema cliente-servidor de altas prestaciones en empresas e instituciones.

1.11 Estado del proyecto:

Debido al crecimiento de las tecnologías de bajo costo que se emplean en dispositivos móviles, que permite implementar nuevos enfoques en diferentes áreas para el manejo de la información. La creación de un sistema cliente-servidor con este tipo de tecnología, permitirá bajar los costos, reducir el consumo de energía y optimizar el espacio en las empresas. El fundamento de este proyecto es crear un servidor de gestor de aplicaciones que permita potencialmente un servicio continuo a los clientes de la red a un costo menor que una infraestructura cliente servidor tradicional.

2. Contenido

2.1 Motivación de la investigación:

RaspberryPi2 es una tecnología que está siendo utilizada recientemente en países como: EE.UU, Inglaterra y otros de Europa con distintas finalidades. El crecimiento de esta tecnología permite implementar servidores a un costo mucho más bajo que los sistemas tradicionales, además posibilita trabajar con diferentes aplicaciones y adaptarlo de acuerdo a las necesidades de las empresas e instituciones.

2.2 Problemática:

La problemática de las empresas es la implementación de tecnología para el manejo de su información debido a su elevado costo, complejidad de la implementación,

tamaño de la infraestructura y al alto consumo energético de los sistemas tradicionales. Esta situación podría dejar de lado a muchas pequeñas y medianas empresas que no pueden manejar dicha complejidad, provocando una brecha tecnológica que es necesario mitigar.

La tendencia de los dispositivos móviles ha dado camino a tecnologías más asequibles con mayores prestaciones a bajo costo. El enfoque principal de esta investigación es conocer más a profundidad la tecnología de los dispositivos *RaspberryPi2*, implementando un sistema cliente-servidor de bajo costo con similares o mejores prestaciones que las tradicionales. Según nuestra indagación previa, al parecer este tipo de tecnología no ha sido empleada en nuestro medio aun, con esta investigación se pretende solucionar el problema tecnológico que muchas empresas tienen.

Los problemas más comunes que presentan estos sectores, es el escaso presupuesto para la implementación de servidores tradicionales por su alto costo. Por lo cual es necesario investigar otras alternativas que brinden las mismas prestaciones que un servidor normal, pero a un costo más bajo.

El desarrollo tecnológico es diverso, de ahí nace el interés de realizar el estudio de este dispositivo, que brindará las mismas prestaciones de un servidor tradicional. Se contribuirá al desarrollo de la pequeña y mediana industria que frecuentemente son quienes presentan más problemas para la adquirir los equipos tradicionales, ya sea por su costo elevado o por las dificultades en la implementación a nivel de software y hardware.

2.3 Resumen:

El propósito es investigar una nueva tecnología y dispositivo denominado *RaspberryPi2*. Con este dispositivo, se desarrollara un sistema cliente-servidor que permitirá un manejo de datos seguros como se lo haría con un servidor tradicional de alto costo. Se trata de demostrar las ventajas, y cuáles son las limitaciones que esta tecnología tiene en lo concerniente a servidores y redes.

RaspberryPi2 brinda todas las facilidades para suplir dichas necesidades dentro del servicio de redes y servidores, gracias a su flexibilidad y adaptabilidad dentro del sector empresarial, la instalación de un sistema cliente-servidor, puede ser implementada en cualquier pequeña o mediana empresa que requiera este servicio. Este sistema, brindará todas las facilidades, dentro del proceso de servicio de cliente- servidor a un costo menor. Su implementación se podría dar en empresas con muy bajo presupuesto, aportando a un proceso de desarrollo y al cumplimiento de sus objetivos organizacionales.

2.4 Indagación exploratoria y base conceptual:

A nivel mundial el incremento de dispositivos nuevos que permiten dar solución a los problemas tecnológicos en las empresas de una manera más simple y económica se ha incrementado de manera considerable, estos dispositivos son mucho más estables y de mayor confiabilidad, dando así mayor seguridad al investigador para su aplicación, razón por la cual se presenta la investigación de este dispositivo *RaspberryPi2*, e implementación de una infraestructura cliente-servidor. *RaspberryPi2* es empleado en diferentes áreas como: Electrónica, Mecánica, Tecnológica, etc. Pero esta investigación se enfoca en el área de servidores y redes debido a las prestaciones que estos dispositivos presentan.

En la actualidad la investigación de nuevos dispositivos tecnológicos en el Ecuador, se ha convertido en base fundamental para el desarrollo del país. La búsqueda de nuevos dispositivos, que permita desarrollar tecnologías, con las mismas o mejores prestaciones a menor costo y mayor funcionalidad, se ha vuelto una constante. Los centros educativos fiscales, así como también las pequeñas empresas, son los sectores más vulnerables y propensos a no contar con dicha tecnología, por lo que no cuentan con un presupuesto pre-establecido, ya sea para cambio de equipos o para su compra, por sus altos costos.

Con la investigación de este nuevo dispositivo, se implementara un sistema cliente-servidor con mayores beneficios a los existentes, y con apertura de nuevas tecnología y servicio de datos, y con ello se procura ayudar e impulsar a que la microempresa se desarrolle, agilitando sus trabajos y tratando de brindar los mismos servicios que las empresas grandes.

El presente estudio se sustenta en las siguientes herramientas.

- Sistemas Open Source
- Tecnología ARMv7
- RaspberryPi2

En el desarrollo de este tema de investigación, se utilizaran herramientas **Open Source,** por su libre accesibilidad y adaptabilidad a los cambios que con lleve esta investigación. Lo que permitirá bajar sus costos aún más en la implementación y desarrollo de nuevos servicios.

El sistema operativo Linux es gratuito, y brinda la posibilidad de trabajar sobre servidores, además, permite la administración de datos de una manera eficaz y confiable.

La protección de datos y el rendimiento para el cliente dentro de la estructura cliente-servidor es de vital importancia, por lo que este sistema operativo es bastante confiable y al alcance de todo tipo de usuario, por su disponibilidad y seguridad y bajo costo que actualmente brinda. [1]

En la actualidad Linux es uno de los sistemas operativos más utilizados, cuenta con una característica única, código abierto (el código abierto da la posibilidad de mejorar lo actualmente obtenido por investigadores, para obtener un software de mejor calidad).

Microsoft SQL Server, es la versión gratuita del popular sistema de gestión de bases de datos de Microsoft. Es un sistema de administración de datos eficaz y confiable que además ofrece sus lenguajes para consultas; T-SQL y ANSI SQL, y un variado conjunto de características, protección de datos y rendimiento para clientes de aplicaciones incrustadas, aplicaciones web y almacenes de datos locales. Está disponible de forma gratuita y su redistribución con aplicaciones también es gratuita, que contribuye eficientemente a este proyecto. [2]

Además, para demostrar la funcionalidad del sistema en lo que se refiere a interfaz del usuario y manipulación de datos, será necesario utilizar una herramienta como Netbeans, que es un programa de desarrollo libre, creado para manejar de una mejor manera la programación estructurada en java, este es una plataforma que da la posibilidad de trabajar con módulos, es decir que permite extender su programación de ser necesario, debido a que cada uno de estos módulos pueden ser desarrollados independientemente uno del otro, es por ello que estos módulos pueden ser extendidas de una forma más fácil y segura por otros desarrolladores.

El desarrollo en Netbeans es multifuncional, ya que cuenta actualmente con una plataforma de desarrollo IDE (J2Se, Web, EJB, y todo tipo de aplicaciones móviles), por lo tanto, esta se convierte en una estructura de integración modular que servirá para la integración de varias aplicaciones tanto de escritorio como móviles,

La implementación del Clúster de servidores mediante *RaspberryPi2,* permitirá el estudio de la ***tecnología ARMv7,*** que es una arquitectura que se utiliza actualmente en los dispositivos móviles existentes de las más altas prestaciones, por lo tanto, dará la posibilidad de tener un conocimiento más amplio de esta tecnología, enfocada claramente en el manejo estructurado de datos.

Todas y cada una de estas herramientas servirán para el en el desarrollo de esta investigación, que tiene como objetivo la implementación de un clúster de servidores mediante el dispositivo *RaspberryPi2.* [3]

RaspberryPi2.- *RaspberryPi2* es una tecnología, que permite realizar diversidad de aplicaciones investigativas, de distinta índole y propósito, en carreras como: telecomunicaciones, electrónica, base de datos, etc. Dentro de este contexto, los investigadores pueden utilizar sus funcionalidades y adaptar este dispositivo para suplir sus necesidades en el área tecnológica que se deseen implementar. Es decir, con la misma placa, se puede crear desde un simple servidor para conectar en red 5 computadoras, hasta crear un sistema cliente-servidor para empresas muy grandes.

RaspberryPi2 es una placa electrónica que tiene un costo muy bajo con relación a las bondades tecnológicas que esta brinda, su implementación es diversa, es decir, se lo puede aplicar en las diversas ramas tecnológicas. [4]

Una característica relevante de *RaspberryPi2*, es la de ser una placa integradora, donde se puede adaptar otros dispositivos, con lo cual se logra una mayor funcionalidad. Aquellas integraciones de varias placas de *RaspberryPi2*, dan como resultado obtener una mayor potencia y una mejora sus prestaciones y bondades, se puede implementar todo tipo de sistema operativo, dependiendo del área en la que se lo quiera utilizar, y de las necesidades que se quiera suplir para el desarrollo de proyectos o aplicaciones. [5]

RaspberryPi2 es un dispositivo que ha sido investigado he implementado en diversas áreas por investigadores, debido a su bajo costo y muy altas prestaciones de desarrollo en proyectos investigativos. [6]

2.5 Objetivo general:

Implementar un sistema cliente-servidor mediante tecnología *RaspberryPI2* que garantice los mismos beneficios de un servidor tradicional a un costo asequible a empresas e instituciones de tipo pymes.

2.6 Objetivos específicos:

1. Sistematizar conceptos y fundamentos de la tecnología *RaspberryPI2*.
2. Desarrollar un sistema cliente–servidor con *RaspberryPI2.*
3. Implementar servicios de redes y base de datos en el clúster
4. Generar pruebas de rendimiento y disponibilidad

2.7 Metodología:

El nivel de investigación será aplicada, ya que se implementara un clúster Cliente-Servidor con la utilización de *RaspberryPI2*, que es un nuevo dispositivo, en lo que se refiere a la creación de servicios de redes y base de datos en clúster,

La implementación y utilización de este dispositivo para la creación de clúster lleva a investigar todo lo concerniente a: revistas, publicaciones, artículos referentes a RaspberryPI2, ya que será de mucha ayuda en la implementación de dicha investigación.

Esta implementación será afirmada en el momento en que se hagan las pruebas de rendimiento y confiabilidad en el sistema, garantizando y cumpliendo nuestro objetivo planteado y garantizando así el éxito de nuestra investigación.

MEDIOS

Las fuentes para la recopilación de información serán las siguientes:

- Bibliografía
- Artículos relacionados publicados en Internet
- Herramientas a utilizar; (todo Open Source).

2.8 Alcances y resultados esperados:

Con la investigación del dispositivo *RaspberryPI2,* se implementará un clúster de servidores, de bajo costo y de mejores prestaciones que los servidores actuales, que permita ser implementado en cualquier empresa o institución que requiera infraestructura de bajo costo.

La Investigación y posterior implementación del dispositivo *RaspberryPI2,* permitirá tener los siguientes resultados.

- **Optimización de Recursos:** Con la implementación de este dispositivo, se permitirá manejar de una manera más segura y eficiente los recursos.
- **Tecnologías ARMv7:** Permite implementar nuevas arquitecturas de desarrollo, implementadas en países de primer mundo como: EE.UU, Inglaterra, Alemania, etc.

- **Bajos Costos:** Con la Implementación de *RaspberryPI2* y la ***optimización de recursos*** que permiten, bajar los costos de implementación de este sistema

- **Desarrollo Empresarial:** Permitirá ayudar a las microempresas así como a las escuelas fiscales a su desarrollo, ya que es accesible por su *bajo costo* y sus altas prestaciones.

- **Clúster de Alta prestaciones:** Crearemos un Clúster que garantizara el servicio igual o mejor que los existentes independientemente de los S.O a utilizar.

2.9 Supuestos y riesgos:

Supuestos	Riesgos	Posible Solución
La investigación se muestra viable, debido a que este dispositivo ya ha sido utilizado en otros países con estos fines, y existe información al respecto sobre su utilización.	La información que existe no sea lo suficientemente consistente en cuanto a problemas de desarrollo o implementación de la investigación.	Utilizar todo tipo de referencias e información confiable existentes.
En el desarrollo de la investigación no existan imprevistos. Y que los plazos y tiempos estimados se den acuerdo a lo planificado.	Se produzcan imprevistos, que modifiquen los plazos y tiempos estimados Ejemplo; daño de la placa.	Evaluar este tipo de inconvenientes, y a su vez incorporarlos en los tiempos y plazos de entrega.
Que todas las aplicaciones sean compatibles.	Imprevistos en los resultados esperados debido a incompatibilidad de aplicaciones.	Investigar nuevas tecnologías que permitan el soporte de nuevas aplicaciones.
El dispositivo	El dispositivo no	Tratar de suplir estas

RaspberryPI2 tenga la potencia y características de desarrollo esperados.	presente las características esperadas	carencias con la implementación de otros dispositivos que suplan estas deficiencias.
El análisis de pruebas del clúster se lo realice de forma correcta.	El clúster no garantice el servicio por algún problema de placa.	Verificar información que ayude aplicar la distribución de cargas en cada placa por supuestos daños, garantizando su servicio.

2.10 Presupuesto:

Para llevar a cabo esta tesis se necesita financiar los siguientes valores:

Descripción	Costo (dólares)
Compra de Raspberry.	$ 80
Teclado	$15
Impresiones	$ 100
Conexión a Internet	$ 90
Imprevistos	$ 50
Transporte	$ 50
Luz, teléfono, etc.	$80
TOTAL	**$ 465**

2.11 Financiamiento:

Para el desarrollo de este proyecto, se está realizando una inversión personal, debido al costo bajo de los dispositivos, permite realizar un gasto bajo. Sin embargo, será presentado a empresas quienes podrán realizar una inversión mayor.

2.12 Esquema tentativo:

1 ESQUEMA DEL CONTENIDO

Abstract

Introducción

Resumen

Objetivos

Capítulo 1: Tecnologías aplicadas

1.1 Introducción

1.2 Dispositivo Raspberrypi2

1.2.1 Procesadores ARMv7

1.2.2 Clúster servidor RaspberryPi2

1.2.3 Sistemas Distribuidos

1.2.4 Seguridad

1.3 Aplicaciones y servicios de red

Capítulo 2: Sistema cliente-servidor

2.1 Servicios cliente-servidor

2.1.1 Introducción

2.1.2 Componentes

2.1.3 Objetivos

Capítulo 3: Desarrollo e implementación.

3.1 Desarrollo del clúster.

3.2 Implementación del clúster.

Capítulo 4: Pruebas de rendimiento y disponibilidad.

Conclusiones

Bibliográfica

Fuentes electrónicas de consulta

2.13 Cronograma:

Tiempo	Mes1				Mes2				Mes3				Mes4			
Actividades	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Elaboracion y aprobacion del Diseño																
Sistematizar conceptos y fundamentos de la tecnología *RaspberryPI2*																
Desarrollar un sistema cliente–servidor con *RaspberryPI2.*																
Implementar servicios de redes y base de datos en el clúster																
Generar pruebas de rendimiento y disponibilidad																
Revisión																
Entrega Final																

2.14 Referencias:

[1] J. Kiepert, «Creating a Raspberry Pi-Based Beowulf Cluster,» *RpiCluster,* p. 30, 2013.

[2] J. Celkos, SQL for Smarties, EE.UU: Elsevier, 2011.

[3] D. Brash, «Extensions to the ARMv7-A Architecture,» ARM, 2010.

[4] Raspberrypi, «MAKE RESOURCES,» Raspberrypi, 2010. [En línea]. Available:

https://www.raspberrypi.org/resources/make/. [Último acceso: 13 10 2015].

[5] Raspberrypi, «PICADEMY,» Raspberrypi, 2010. [En línea]. Available: https://www.raspberrypi.org/picademy/. [Último acceso: 13 10 2015].

[6] Raspberrypi, «TEACH RESOURCES,» Raspberrypi, 2010. [En línea]. Available: https://www.raspberrypi.org/resources/teach/. [Último acceso: 13 10 2015].

[7] J. T. C. R. P. B. S. J. J. w. Simon J. Cox, «Iridis-pi: a low-cost, compact demonstration cluster,» *Cluster Computing,* vol. 17, nº 2, p. 358, 2013.

[8] D. White, S. Jouet, J. Singer y D. Pezaros, «The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures,» *IEEE Xplore Digital Library,* p. 112, 2013.

[9] n.d, «Raspberry Pi,» *sheffieldlearningcommunity,* p. 1045, 2013.

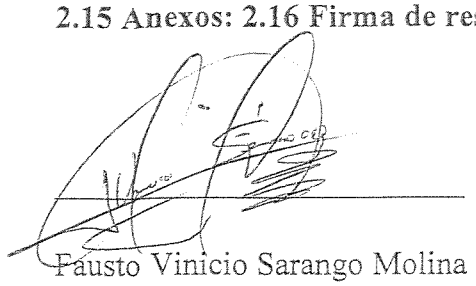
[10] C. G. Crespo, «Diseño y evaluación de un clúster HPC: Aplicaciones,» *UPC,* p. 80, 2014.

[11] P. Chan, «Raspberry Pi User Guide,» element14, Ingaterra, 2015.

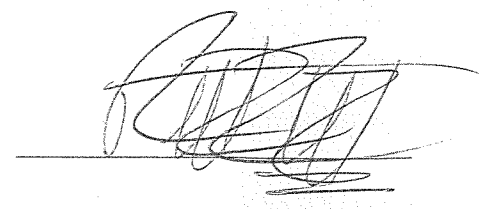
[12] raspberrypi, «WHAT IS A RASPBERRY PI?,» raspberrypi, 2010. [En línea]. Available: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/. [Último acceso: 14 10 2015].

[13] Rraspberrypi, «LEARN RESOURCES,» Rraspberrypi, 2010. [En línea]. Available: https://www.raspberrypi.org/resources/learn/. [Último acceso: 13 10 2015].

2.15 Anexos: 2.16 Firma de responsabilidad

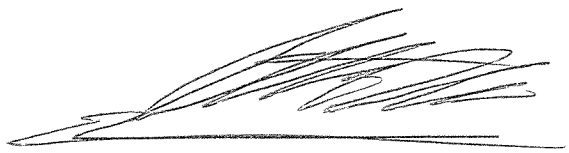


Fausto Vinicio Sarango Molina



Rubén Omar Riofrío Ocaña

2.17 Firma de responsabilidad



Ing, Marco Orellana

2.18 Fecha de entrega: