



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería en Electricidad y Computación**

“DISEÑO DE UN CLÚSTER PARA LA IMPLEMENTACIÓN DE  
SERVICIOS DE RED UTILIZANDO HARDWARE DE BAJO  
COSTO”

**INFORME DE MATERIA INTEGRADORA**

Previa a la obtención del Título de:

**LICENCIADO EN REDES Y SISTEMAS OPERATIVOS**

ANGELO SALVADOR BURGOS ZORRILLA  
FRANCISCO JAVIER PROAÑO NAVARRETE

GUAYAQUIL – ECUADOR

AÑO: 2016



## **AGRADECIMIENTOS**

Mi agradecimiento en primera instancia es a Dios, pues sin Él nada hubiera podido ser posible en mi vida. Indudablemente agradecer a mis padres y mis hermanas es lo procedente pues sin su apoyo no hubiera podido llegar a culminar esta etapa de mi vida; ellos han sido copartícipes de todas mis decisiones y sin importar los resultados jamás han dejado de creer en mí y me han acompañado en cada paso que he dado. Así también es meritorio mencionar a mi tía Edita fiel acompañante de este largo camino recorrido. A todas gracias por su confianza, su apoyo y por infundirme fortaleza en todo momento.

*Francisco Javier Proaño Navarrete*

Este trabajo lo dedico a Dios por darnos la sabiduría, la inteligencia y por darnos los medios suficientes para llegar a este punto de mi carrera. A mi familia, por su incondicional apoyo tanto en la parte afectiva como en la parte económica.

*Angelo Salvador Burgos Zorrilla*

## DEDICATORIA

Dedico este trabajo a Dios y a mi familia, pero especialmente a mi madre, quien con su ejemplo supo inculcarme altos valores como la fortaleza, la constancia, la honestidad. Valores que los he aplicado a cada etapa de mi vida. A ti mamá dedico este trabajo símbolo de mi reconocimiento a tu esfuerzo y a tu amor incondicional que me ayudaron a ver más allá de lo que yo podía ver. Y aunque no esté presente le dedico este trabajo a mi padre de quien orgulloso llevo su nombre y de quien siempre recibí los mejores ejemplos.

*Francisco Javier Proaño Navarrete*

Agradezco a Dios, a mi familia, profesores; por impartirme sus conocimientos y experiencias.

*Angelo Salvador Burgos Zorrilla*

## TRIBUNAL DE EVALUACIÓN

---

**Ing. José Roberto Patiño Sánchez**

PROFESOR EVALUADOR

---

**Msc. Jorge Magallanes Borbor**

PROFESOR EVALUADOR

## **DECLARACIÓN EXPRESA**

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me (nos) corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

---

Angelo Salvador Burgos Zorrilla

---

Francisco Javier Proaño Navarrete

## RESUMEN

Dado que en los últimos años se ha acelerado el avance de la arquitectura en los microprocesadores, ha permitido el desarrollo de la computación en paralelo. Un computador paralelo es un conjunto de procesadores capaces de cooperar en la solución de un problema. El desarrollo del clúster usando Raspberry Pi 2 está pensado para ser implementado en PYMES que quieren usar las características de este dispositivo de bajo costo y gran potencial. Y con esto lograr tener una plataforma alterna a los grandes y costosos sistemas de supercómputo.

Nuestro clúster mantiene un estado de alto procesamiento; esto quiere decir que usa todos los recursos como: memoria RAM, procesador, espacio en el micro sd que usa como si fuera un disco duro, puertos en general. El diseño del chasis fue diseñado con un botiquín para transportar el clúster vpn/web y el clúster owncloud y evitar cualquier daño; nuestra arquitectura mantiene un entorno de alto rendimiento para aprovechar los equipos Raspberry Pi conectados a cada subnet del clúster y bajar el tiempo de respuesta a los procesos.

Al hacer comparativas de los diferentes tipos de Raspberry Pi en cada clúster se puede apreciar que cada tipo tendrá una desventaja contra el más avanzado; aunque mantiene la compatibilidad y su uso dentro del clúster, también una opción de fusionar todos los nodos y crear un clúster único que no fue nuestro caso para mostrar la diferencia que existe en los tipos y realizar las pruebas necesarios del funcionamiento y proceso.

## ÍNDICE GENERAL

AGRADECIMIENTOS .....	ii
DEDICATORIA.....	iii
TRIBUNAL DE EVALUACIÓN .....	iv
DECLARACIÓN EXPRESA .....	v
RESUMEN .....	vi
ÍNDICE GENERAL.....	vii
ÍNDICE DE TABLAS .....	x
ÍNDICE DE FIGURAS .....	xi
CAPÍTULO 1 .....	1
1. ALCANCE Y PROBLEMÁTICA .....	1
1.1 Situación Actual.....	3
1.2 Propósito .....	4
1.3 Justificación .....	5
1.3.1 Justificación de la arquitectura .....	6
1.4 Objetivos .....	7
1.4.1 Objetivos General.....	7
1.4.2 Objetivos Específicos .....	7
1.5 Alcance.....	7
1.6 Metodología.....	8
1.6.1. Adquisición del hardware.....	8
1.6.2. Selección y obtención del software .....	9
1.6.2. Instalación, configuración .....	9
1.7 Requerimientos básicos .....	9
2. FUNDAMENTOS TEÓRICOS. ....	11
2.1 Raspberry Pi.....	11
2.2 Características técnicas del Raspberry Pi .....	11
2.3 Modelos.....	12



2.3.1.	Raspberry Pi Modelo A.....	12
2.3.2.	Raspberry Pi Modelo A+.....	14
2.3.3.	Raspberry Pi Modelo B.....	16
2.3.4.	Raspberry Modelo B+.....	18
2.3.5.	Raspberry Pi 2 Modelo B.....	20
2.3.6.	Raspberry Modelo Zero.....	22
2.4.	Procesador ARM.....	24
2.4.1.	Arquitectura ARM.....	24
2.5	Instaladores y Sistemas Operativos para Raspberry Pi. ....	25
2.5.1.	Instaladores:.....	25
2.5.2.	Sistemas Operativos:.....	26
2.6	Instalador NOOBS.....	27
2.7	Característica del Sistema Operativo Raspbian. ....	27
2.8	Competitividad en el mercado. ....	30
2.8.1.	Cubeboard.....	30
2.8.2.	Banana Pi.....	32
2.8.3.	ODROID U3.....	33
2.9	Computación Paralela.....	35
2.10.	Fundamentos Generales de un Clúster.....	36
2.10.1.	Concepto.....	36
2.10.2.	Características.....	37
2.10.3.	Clasificación.....	37
2.11.	Implementando Linux como Sistema Operativo.....	38
2.12.	Definición de Servidor.....	39
2.12.1.	Tipos de Servidores.....	39
2.13.	Proyecto Hadoop.....	40
3.	CONFIGURACIÓN DE CLÚSTERS. ....	41
3.1.	Diseño de un clúster utilizando plataforma Raspberry Pi.....	41

3.2. Diseño del primer clúster con 3 Raspberry Pi B+ .....	44
3.3. Requerimientos de hardware y software .....	45
3.4. Implementación del primer clúster.....	46
3.5. Introducción a MPI.....	46
3.6. Protocolo que permite conexiones remotas seguras SSH. ....	48
3.7. Uso de herramienta NMAP (NETWORK MAPPER). ....	49
3.8. Instalación y configuración del Sistema Operativo .....	50
3.9. Instalación del servicio VPN (VIRTUAL PRIVATE NETWORK) .....	51
3.10 Diseño del segundo clúster con 3 Raspberry Pi 2+ .....	51
CONCLUSIONES Y RECOMENDACIONES .....	53
BIBLIOGRAFÍA .....	55
ANEXOS.....	58
Despliegue del plan de trabajo (diagrama de gantZ) .....	59

## ÍNDICE DE TABLAS

Tabla 1: Materiales a utilizar .....	9
Tabla 2: Características Raspberry Pi Modelo A.....	13
Tabla 3: Características Raspberry Pi Modelo A+.....	15
Tabla 4: Características Raspberry Pi Modelo.....	17
Tabla 5: Características Raspberry Pi Modelo B+.....	19
Tabla 6: Características Raspberry Pi 2 Modelo B.....	21
Tabla 7: Índice de Refracción de Materiales.....	23
Tabla 8: Características del CUBIEBOARD.....	31
Tabla 9: Características BANANA PI.....	32
Tabla 10: Características de ODROID-U3.....	34
Tabla 11: Costo de materiales.....	35
Tabla 12: Niveles de computación paralela.....	36
Tabla 13: Características del primer nodo del clúster con Raspberry PI B+.....	41
Tabla 14: Características del segundo nodo del clúster con Raspberry PI B+.....	42
Tabla 15: Características del tercer nodo del clúster con Raspberry PI B+.....	42
Tabla 16: Características de conmutador TREDnet.....	43
Tabla 17: Características de conmutador D-link.....	44
Tabla 18: Pruebas en clúster.....	52

## ÍNDICE DE FIGURAS

Figura 1.1: Situación actual de la empresa. ....	4
Figura 1.2: Propuesta de trabajo.....	5
Figura 2.1: Raspberry Pi Modelo A. [17] .....	14
Figura 2.2: Raspberry Pi Modelo A+. [18] .....	16
Figura 2.3: Raspberry Pi Modelo B. [17] .....	18
Figura 2.4: Raspberry Pi Modelo B+. [19] .....	20
Figura 2.5: Raspberry Pi 2 Modelo B. [20] .....	22
Figura 2.6: Reflexión de un rayo de luz. [21].....	24
Figura 2.7: Arquitectura Del Raspberry.....	25
Figura 2.8: CUBIEBOARD. [22].....	32
Figura 2.9: BANANA PI. [8] .....	33
Figura 2.10: ODROID-U3. ....	34
Figura 3.1: Diseño del primer clúster.....	42
Figura 3.2: Clúster con tre raspberry Pi B+ .....	44
Figura 3.3: Ejemplo de envío de información utilizando MPI .....	45
Figura 3.4: Gráfica de protocolo SSH.....	46
Figura 3.5: Funcionamiento de escaneo con herramienta NMAP .....	47
Figura 3.6: Arquitectura de un servidor VPN.....	48
Figura 3.7: Arquitectura de un servidor WEB .....	49

## CAPÍTULO 1

### 1. ALCANCE Y PROBLEMÁTICA

Es innegable el crecimiento que ha tenido la economía del Ecuador, su estructura económica se ha afianzado en la creación de micro, pequeñas y medianas empresas, así como en la expansión de las grandes industrias del país.

En este contexto podemos acotar que el papel que desempeñan principalmente las pequeñas y medianas empresas (PYMES) es importante dentro de la dinámica económica de un país que concentra su actividad comercial en un 72,6% en las dos provincias principales Guayas y Pichincha [1].

De acuerdo con lo expuesto por Christian Cisneros, director ejecutivo de la Cámara de la Pequeña y Mediana Empresa de Pichincha (Capeipi), las mipymes representan el 99% del tejido empresarial del país; son el motor de la economía indica, ya que sus cifras de ventas por año ascienden hasta \$4M con una nómina de hasta 200 trabajadores. [2]

Sin embargo, este vertiginoso crecimiento de las PYMES tiene su talón de Aquiles, en cuanto a la tecnificación de su actividad comercial tanto en cuanto al marketing a través de redes sociales (hoy es imprescindible para el crecimiento de cualquier negocio) como en la eficiencia de sus procesos y mejora de la productividad.

Podríamos identificar dos grandes razones para esta situación, la primera el desconocimiento de las nuevas tecnologías que pueden aplicarse a sus empresas y la segunda la presunción de que toda solución tecnológica representa un alto costo por los precios a los que se comercializan los equipos de computación en nuestro país.

Ciertamente la inversión en tecnología en las PYMES no es el rubro principal dentro de la inversión presupuestada ya que dada las actuales condiciones en el país para la importación de equipos hace que al momento de incursionar en

la búsqueda de mejores tecnologías las cifras se disparen haciendo que esto pase a un último escaño dentro del plan de crecimiento del pequeño y mediano empresario.

Es importante contar entonces con una alternativa que permita al empresario mejorar su estructura de información y tecnología sin que esto represente un elevado costo dentro de su presupuesto. Que no solo sea una solución temporal a la necesidad de almacenar información, sino que además sea escalable y convergente de tal forma que a medida que su negocio crezca pueda adaptar nuevos nodos según su necesidad.

Esta alternativa deberá responder a los dos grandes retos de la economía moderna. El primero, ser económicamente competitiva y la segunda de fácil acceso. No sería una alternativa viable si no es asequible al mercado ecuatoriano o si para adquirirla además de los costos de instalación habría que adicionar costos por importación además de la dificultad que esto sugiere al empresario que no cuenta con el conocimiento necesario para importar estos equipos.

En este contexto se hace imperativo contar en el mercado (considerando a las pymes como el nicho apetecido) con una solución que proporcione los mismos o mejores beneficios que los equipos tradicionales, pero a un menor costo de instalación y mantenimiento, que sea escalable, que sea compatible con los programas que actualmente demandan las empresas y que se comercialicen en el mercado local.

Nuestro proyecto lo trabajamos con Raspberry Pi 2, porque es una herramienta tecnológica que poco a poco ha ido tomando más fuerza para el desarrollo de proyectos gracias a su bajo costo y su gran potencial. El desarrollo del Clúster conformado por Raspberry Pi 2 está pensado para ser implementado en PYMES que quieran aprovechar las grandes características de esta placa.

Hemos hecho un análisis a un estudio jurídico donde hemos encontrado muchas falencias a nivel tecnológico; viendo este panorama hemos propuesto nuestra solución de desarrollar un Clúster con Raspberry Pi 2 y utilizando

software libre. A continuación, daremos a conocer el estado actual del estudio jurídico y de cómo puede mejorar con la implementación de nuestro proyecto.

Silva y Asociados es un reconocido estudio jurídico a nivel de Ecuador, que ofrece servicios tales como recuperación de automóviles, juicios laborales, declaración de bienes e inmuebles, asesoría legal en litigios. Tiene su matriz en Quito, dos oficinas en el edificio Amazonas que se encuentra en pleno centro de Quito que cuenta con 30 empleados. En la matriz es donde se manejan todas las carteras de recuperación de vehículo.

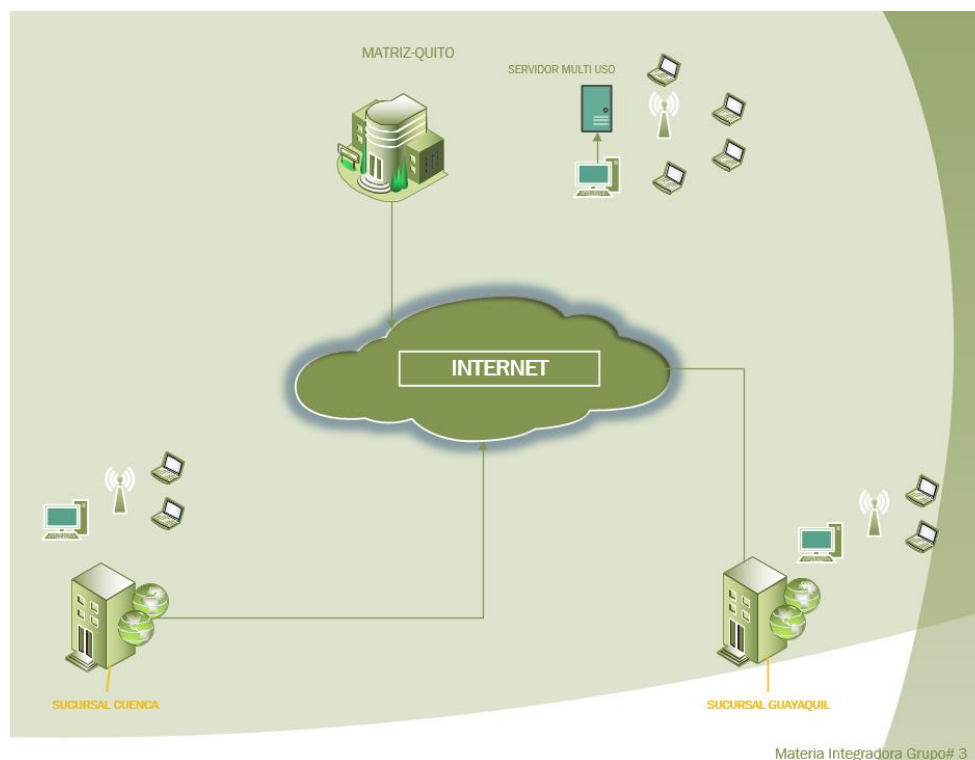
La sucursal Cuenca, que está ubicada en el norte de la ciudad y cuenta con 10 trabajadores. La sucursal de Guayaquil se encuentra ubicada en Las Torres del Norte en la ciudadela Urdesa Norte que posee 8 empleados y por último posee una pequeña oficina en Manta ubicada junto al Puerto Marítimo y cuenta con 3 trabajadores.

## **1.1 Situación Actual**

Silva y Asociados han incrementado su tamaño, su volumen de trabajo y está pensando en contratar 10 empleados más y espera extenderse a otras ciudades. La empresa desea brindar un excelente servicio a sus clientes, pero también necesita tener un control de sus gastos en lo que se refiere a infraestructura informática, por lo que requiere optimizar dicha infraestructura y aumentar su rentabilidad lo que implica reducir los gastos operativos.

Dado que la empresa ha incrementado su capital debido al apoyo de sus socios, tomo la decisión de adquirir una plataforma tecnológica que permitirá dar asesoría jurídica mediante una página web.

La empresa posee un pequeño servidor en la matriz, el que es utilizado únicamente para almacenar instaladores de las impresoras que se encuentran en red. La comunicación entre las sucursales es únicamente por correos electrónicos.



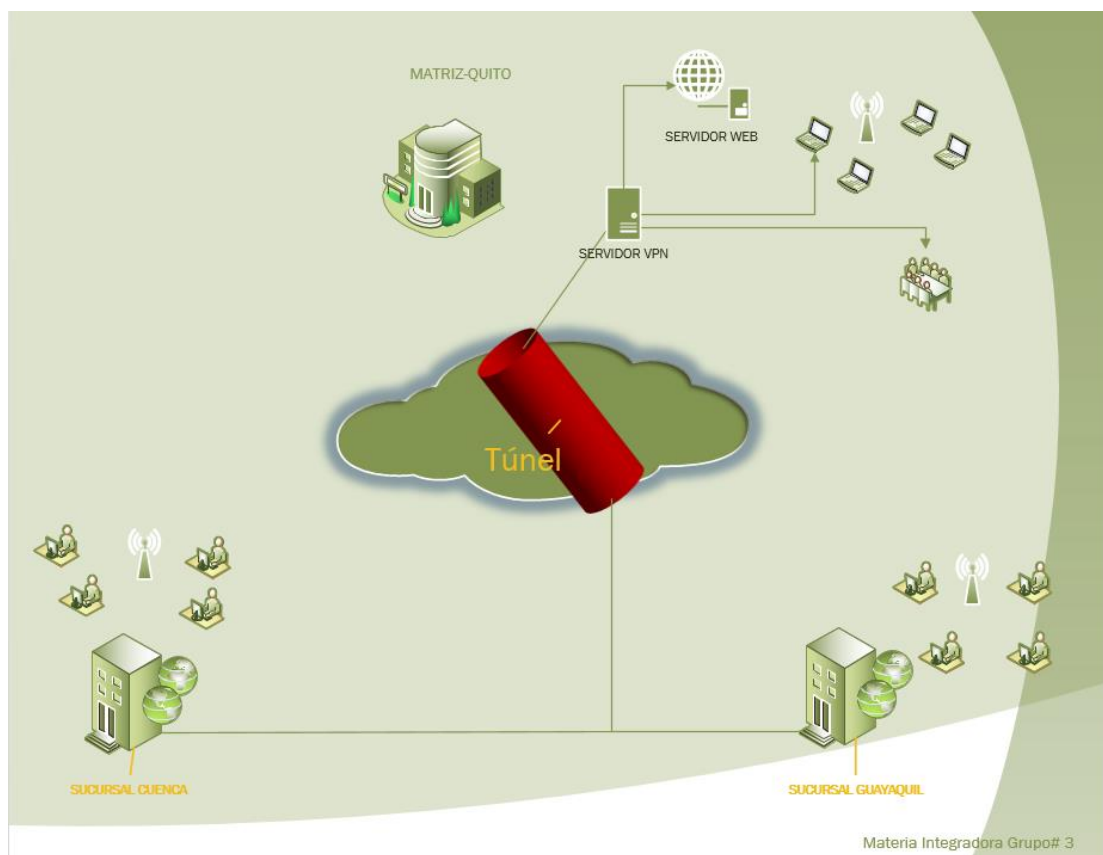
**Figura 1.1: Situación actual de la empresa.**

## 1.2 Propósito

La construcción de un clúster nos posibilita la implementación de un mecanismo de redundancia que permita que los servicios se encuentren disponibles la mayor cantidad de tiempo para que los usuarios puedan realizar sus trámites en el momento que ellos lo deseen.

Creación de un servidor VPN que me garantice la interconexión de la red de forma segura a través de una infraestructura pública, acceso remoto a la información de la empresa, entre otras. Implementar un servidor web que nos permita brindar un excelente servicio para el nuevo servicio de asesoría legal por medio de la página web.





**Figura 1.2: Propuesta de trabajo.**

### 1.3 Justificación

Debido al nuevo proyecto en el que ha incursionado la empresa y al incremento en la navegación Web, correo electrónico y transferencia de archivos, se ha visto en la necesidad implementar un mecanismo de redundancia que proporcione la disponibilidad de la plataforma 24 horas al día, los siete días a la semana. Además, cubrir las necesidades de transmitir los datos a través de un servidor VPN de una forma segura, rápida y menos costosa.

Es por esta razón, que se justifica el desarrollo de este proyecto debido a las falencias que posee la empresa a nivel de su infraestructura

informática y las necesidades de redundancia, alta disponibilidad, y seguridad en el intercambio de información.

### **1.3.1 Justificación de la arquitectura**

Una mayor innovación se puede lograr al mantener un servicio con hardware de bajo costo (Raspberry Pi). Este dispositivo tiene un nivel de procesamiento cercano al de un computador y realiza la misma clase de trabajo. Lo interesante es poder contar un entorno realista de computación distribuida, aunque su potencia no la sitúe a la altura de los superordenadores, pero si tiene la capacidad de montar una variedad de servicios como servidores, nubes para back up, sistemas de seguridad, domótica, firewall, entre otros. Lo destacable del proyecto es dar a conocer este hardware de bajo costo y fomentar su uso.

Hemos decidido utilizar Linux debido a su rapidez, su estabilidad y porque es la mejor opción para servidores y aplicaciones distribuidas. También podemos agregar que con un bajo presupuesto podemos obtener el sistema operativo con un sin número de aplicaciones y una gran diversidad de distribuciones que podremos elegir. Otra de las grandes virtudes de las distribuciones de Linux es su compatibilidad con otros sistemas operativos como por ejemplo Windows o Mac OS.

## **1.4 Objetivos**

### **1.4.1 Objetivos General**

Proveer un diseño para la implementación de un clúster de alta disponibilidad implementando Raspberry Pi 2 y utilizando software de distribución libre bajo la licencia GNU/Linux.

### **1.4.2 Objetivos Específicos**

- Proponer una alternativa de bajo costo para implementar un Clúster de Alta disponibilidad.
- Diseñar un modelo de un Clúster sobre la plataforma Raspberry Pi 2 usando un software de distribución libre bajo licencia GNU/Linux.
- Configurar e Integrar seis Raspberry Pi 2 a dos Clústers de Alta disponibilidad con 2 servicios (VPN y WEB).
- Determinar los requerimientos tanto en hardware como en software que serán necesarios para la implementación de la propuesta.
- Analizar de rendimiento del Clúster de Raspberry Pi utilizando la herramienta Sysbench, para determinar su viabilidad.
- Elaborar una serie de directrices para la instalación y configuración del Clúster.

## **1.5 Alcance**

Este proyecto tiene como finalidad proponer una alternativa diferente para la implementación de un Clúster de Alta disponibilidad utilizando Raspberry Pi 2 mediante el empleo de herramientas y aplicaciones de

distribución libre para el Sistema Operativo Linux, y así mantener la interconexión total de la red de forma segura y acceso remoto a la información de la empresa.

Se estudiará la configuración de un Clúster de 3 Raspberry Pi2, se analizará la operatividad y eficiencia del Clúster.

## **1.6 Metodología**

La estrategia que se usará para la implementación de un Clúster de Alta disponibilidad utilizando Raspberry Pi 2 se basará en material publicado en internet tal como: libros electrónicos, blog, páginas web y foros de difusión.

Para llevar a cabo la propuesta planteada en nuestro proyecto se requiere llevar a cabo las siguientes tareas:

- Adquisición del hardware.
- Selección y obtención del software.
- Instalación, configuración y
- Pruebas de funcionamiento.

### **1.6.1. Adquisición del hardware**

La adquisición del hardware se realizará mediante la compañía de comercio electrónico Amazon la que nos proporcionará los Raspberry Pi 2 debido que en el país no se cuenta con esa tecnología o si se llega a encontrar los costos son muy elevados y en la mayoría de los casos son versiones antiguas que no ayudaría a alcanzar las expectativas que se tienen trazadas para este proyecto. El resto de materiales para el desarrollo del proyecto tales como: cables, router, switch, monitor con estrada HDMI, hub de USB, serán adquiridos en COMPUTRON.

### 1.6.2. Selección y obtención del software

Para poder seleccionar el mejor sistema operativo para configurar el Clúster recabaremos información de múltiples foros donde obtendremos referencias acerca de los distintos softwares existentes. Unas de las cosas que tenemos que tener en cuenta antes de la selección del sistema operativo son las siguientes:

- Que la distribución sea estable.
- Recursos necesarios para la instalación.
- Que exista soporte.
- Referencias de tercera persona.

En la obtención del software libre tenemos múltiples formas, una de ellas es por medio de Internet otras formas es a través de medios físicos ya sea memoria flash, DVD o CD.

### 1.6.2. Instalación, configuración

En lo concerniente a la instalación y configuración del sistema operativo se debe leer la documentación, cumplir con los requisitos de hardware y de software.

## 1.7 Requerimientos básicos

Tabla 1: Materiales a utilizar

MATERIAL	CANTIDAD
----------	----------

Raspberry Pi 2	3
Router	1
Switch	1
Cables con conector RJ-45	8
Mouse	1
Teclado	1
Monitor con entrada HDM	1

## CAPÍTULO 2

### 2. FUNDAMENTOS TEÓRICOS.

#### 2.1 Raspberry Pi

Entenderemos como Raspberry Pi a una placa de computadora que está basada en el microcontrolador Atmel ATmega644 de un costo muy bajo. Este dispositivo se lo creo en Reino Unido por la fundación Raspberry PI con el afán de diseñar un pequeño computador que fuera muy económico y que permita a los estudiantes programar y practicar en computadoras sin temor a dañarlas.

El Raspberry Pi es como un ordenador que carece de accesorios y no por esto afecta su funcionamiento básico. [3]

#### 2.2 Características técnicas del Raspberry Pi

Raspberry Pi es una pequeña maravilla que esta comprimida en una tarjeta no más grande que una tarjeta de crédito, pero puede realizar múltiples operaciones.

En el centro de la tarjeta del Raspberry Pi encontramos un semiconductor cuadrado, su nombre es circuito integrado o chip. Este es el Broadcom BCM2835 system-on-chip (SoC). Otro semiconductor del Raspberry Pi es la memoria que almacena temporalmente los datos mientras se corren los programas. El tipo de memoria que usa es conocido como memoria de acceso aleatorio (RAM), ya que el computador puede leer y escribir en cualquier parte de la memoria. RAM es volátil lo que implica que la información que se encuentra almacenada se perderá si Raspberry Pi se queda sin energía o se apaga.

Por otro lado, también están los conectores de salida de video, este es el puerto High Definition Multimedia Interface (HDMI), El conector amarillo está diseñado para conectar con modelos antiguos de televisores que no poseen puerto HDMI.

Contamos en la parte superior izquierda el (GPIO) general-purpose input-output, que es usado para conectar el Raspberry Pi con otro hardware.

Otro conector es el Display Serial Interface (DSI) este puerto se utiliza para conectar el sistema de pantalla digital. El puerto (CSI) Camera Serial Interface nos permite conectar con un sistema de cámara. Posee un conector que permite el paso de electricidad micro-USB. Asimismo, tiene una ranura para la memoria (SD) Secure Digital, esta memoria no es volátil ya que retiene información cuando el Raspberry no dispone de energía. [3]

Podemos encontrar el Raspberry Pi en dos diferentes modelos tales como Modelo A y Modelo B. Una de las diferencias entre estos dos es que el Modelo A sacrifica muchas de sus funcionalidades para así poder reducir su costo.

## **2.3 Modelos**

### **2.3.1. Raspberry Pi Modelo A**

Este modelo pese a su bajo costo, no deja de ser tan poderoso que los otros modelos, la diferencia es que posee menos memoria, 256 MB, las aplicaciones se ejecutan de forma más lenta, este modelo cuenta con un solo puerto (USB) Universal Serial Bus, esto puede complicar las cosas cuando quieras conectar más de un dispositivo por USB al mismo tiempo, se necesitara un USB hub para solucionar este problema.



**Tabla 2: Características Raspberry Pi Modelo A.**

<b>MODELO A</b>	<b>CARACTERÍSTICAS</b>
SoC (chip)	Broadcom BCM2835
Plataforma de hardware	Linux
Sistema Operativo	Linux
CPU	ARM11 ARMv6 700 MHz.
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
Memoria RAM	256 MB LPDDR SDRAM 400 Mhz.
USB 2.0	1
Entrada de video	Cámara CSI
Salida de Video	RCA, HDMI 1.4 @ 1920x1200 pixeles
Salida de Audio	Jack, HDMI
Consumo	300mA / 1.5w / 5v
Alimentación	MicroUSB / GPIO
Tarjetas de almacenamiento	SD/MMC
Ethernet	No
Interfaz periférica(GPIO)	8
GPIO	26-Pin
Tamaño	85,6 x 53,98 mm
Precio actual(Amazon.com)	\$ 25

## RASPBERRY PI MODEL A - \$25

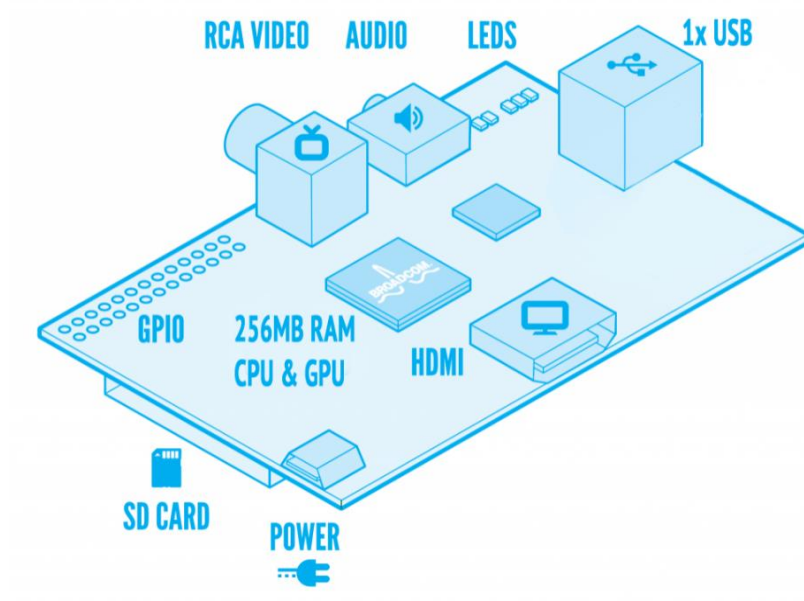


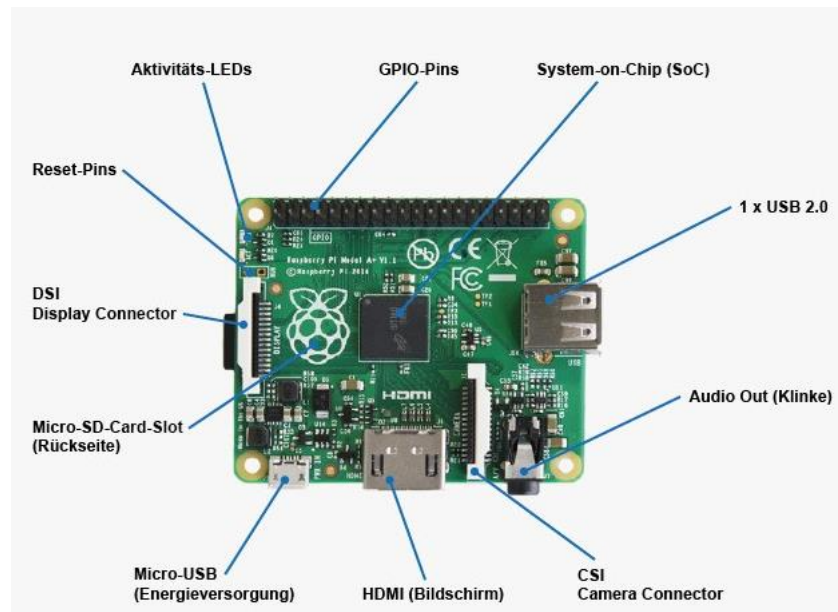
Figura 2.1: Raspberry Pi Modelo A. [17]

### 2.3.2. Raspberry Pi Modelo A+

Este modelo es más pequeño, con mejoras en el hardware como: la nueva ranura Micro SD que funciona con el mecanismo de empujar la tarjeta al momento de introducirla y para extraerla, audio mejorado ya que integra una fuente de alimentación de bajo ruido y lo más destacable es su bajo costo

**Tabla 3: Características Raspberry Pi Modelo A+.**

<b>MODELO A+</b>	<b>CARACTERÍSTICAS</b>
SoC (chip)	Broadcom BCM2835
Plataforma de hardware	Linux
Sistema Operativo	Linux
CPU	ARM11 ARMv6 700 MHz.
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
Memoria RAM	256 MB LPDDR SDRAM 400 MhZ.
USB 2.0	1
Entrada de video	Cámara CSI
Salida de Video	TRRS, HDMI 1.4 @ 1920x1200 pixeles
Salida de Audio	Jack, HDMI
Consumo	400mA / 2w / 5v
Alimentación	MicroUSB / GPIO
Tarjetas de almacenamiento	Micro/SD
Ethernet	No
Interfaz periférica(GPIO)	17
GPIO	40-Pin
Tamaño	65 x 56.5 mm
Precio actual(Amazon.com)	\$ 29.95



**Figura 2.2: Raspberry Pi Modelo A+. [18]**

### 2.3.3. Raspberry Pi Modelo B

Este modelo tiene un precio más elevado que el Modelo A, pero ese viene con ventajas considerable. Internamente incluye un par de memorias de 512MB, mientras que externamente encontraremos puertos adicionales, puede tener dos o cuatro puertos USB, un puerto Ethernet para conectar al Raspberry Pi a la red para tener acceso a internet.

**Tabla 4: Características Raspberry Pi Modelo.**

<b>MODELO B</b>	<b>CARACTERÍSTICAS</b>
SoC (chip)	Broadcom BCM2835
Plataforma de hardware	Linux
Sistema Operativo	Linux
CPU	ARM11 ARMv6 700 MHz.
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
Memoria RAM	512 MB LPDDR SDRAM 400 Mhz.
USB 2.0	2
Entrada de video	Cámara CSI
Salida de Video	TRRS, HDMI 1.4 @ 1920x1200 píxeles
Salida de Audio	Jack, HDMI
Consumo	700mA / 3.5w / 5v
Alimentación	MicroUSB / GPIO
Tarjetas de almacenamiento	SD/MMC
Ethernet	Si, 10/100 Mbps
Interfaz periférica(GPIO)	8
Tamaño	85,6 x 56,5 mm
Precio actual(Amazon.com)	\$ 49.98

## RASPBERRY PI MODEL B

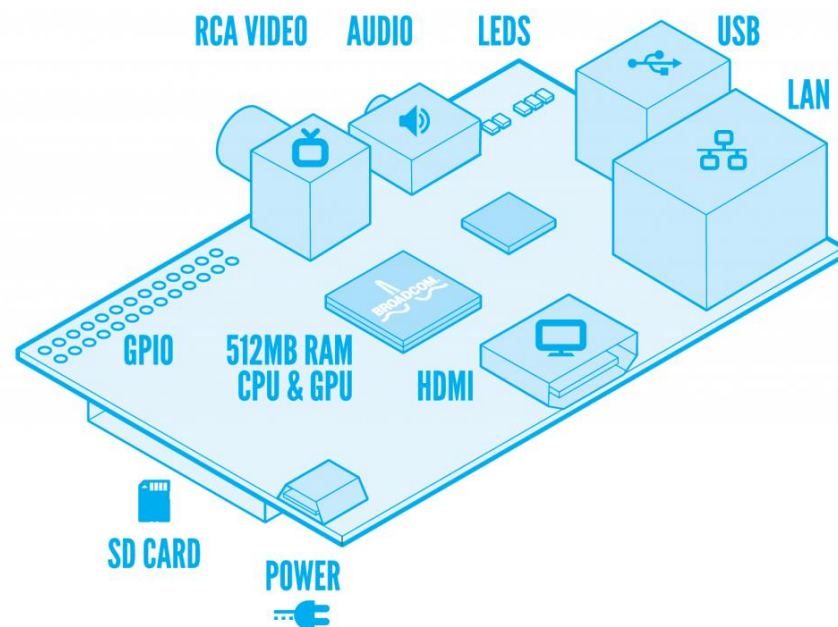


Figura 2.3: Raspberry Pi Modelo B. [17]

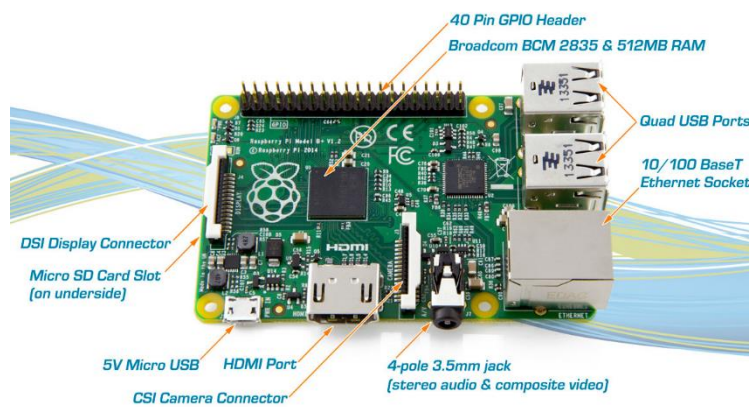
### 2.3.4. Raspberry Modelo B+

Es la última versión desarrollada por la Fundación Raspberry Pi, incluye el mismo procesador CM2835, pero hay grandes cambios como en el conector GPIO con 40 pines, cuatro puertos USB. Este modelo también usa una tarjeta micro-SD. El Modelo B+ es muy interesante si deseamos desarrollar complejos proyectos electrónicos.

**Tabla 5: Características Raspberry Pi Modelo B+.**

<b>MODELO B+</b>	<b>CARACTERÍSTICAS</b>
SoC (chip)	Broadcom BCM2835
Plataforma de hardware	Linux
Sistema Operativo	Linux
CPU	ARM11 ARMv6 700 MHz.
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
Memoria RAM	256 MB LPDDR SDRAM 400 Mhz.
USB 2.0	4
Entrada de video	Cámara CSI
Salida de Video	RCA, HDMI 1.4 @ 1920x1200 pixeles
Salida de Audio	Jack, HDMI
Consumo	500mA / 2.5w / 5v
Alimentación	MicroUSB / GPIO
Tarjetas de almacenamiento	MicroSD
Ethernet	Si, 10/100 Mbps
Interfaz periférica(GPIO)	17
Tamaño	85,6 x 56,5 mm
Peso	45 g
Precio inicial	\$ 35

Precio actual(Amazon.com)	\$ 32.18
---------------------------	----------



**Figura 2.4: Raspberry Pi Modelo B+. [19]**

### 2.3.5. Raspberry Pi 2 Modelo B

La segunda generación de los Raspberry, aproximadamente seis veces más rápida que los modelos anteriores gracias a su memoria RAM de 1 Gb.



**Tabla 6: Características Raspberry Pi 2 Modelo B.**

<b>PI 2 MODELO B</b>	<b>CARACTERÍSTICAS</b>
SoC (chip)	Broadcom BCM2836
Plataforma de hardware	Linux
Sistema Operativo	Linux
CPU	ARM11 ARMv7 ARM cortex-A7 4 núcleos @ 900 MHz.
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
Memoria RAM	1 GB LPDDR2 SDRAM 450 Mhz.
USB 2.0	4
Entrada de video	Cámara CSI
Salida de Video	TRRS, HDMI 1.4 @ 1920x1200 pixeles
Salida de Audio	Jack, HDMI
Consumo	800mA / 4w / 5v
Alimentación	MicroUSB / GPIO
Tarjetas de almacenamiento	microSD
Ethernet	Si, 10/100 Mbps
Interfaz periférica(GPIO)	8
Tamaño	85,6 x 56.5 mm
Precio actual(Amazon.com)	\$ 38.46

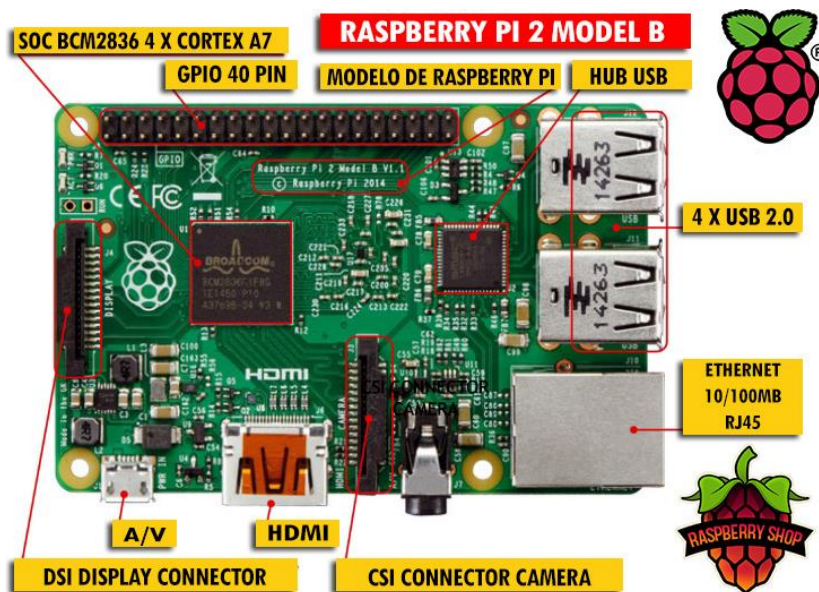


Figura 2.5: Raspberry Pi 2 Modelo B. [20]

### 2.3.6. Raspberry Modelo Zero

Es la última versión desarrollada por la Fundación Raspberry Pi, con un costo de 5 dólares, cuenta con un procesador Broadcom BCM2835 y trabaja a 1 GHz, además cuenta con 512 MB de memoria RAM, una ranura microSD y puerto miniHDMI.

**Tabla 7: Índice de Refracción de Materiales.**

<b>MODELO ZERO</b>	<b>CARACTERÍSTICAS</b>
SoC (chip)	Broadcom BCM2835
Plataforma de hardware	Linux
Sistema Operativo	Linux
CPU	ARM11 a 1GHz
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
Memoria RAM	256 MB LPDDR SDRAM 400 MhZ.
Salida de Video	Mini HDMI 1080p
Alimentación	2 conectores micro-USB
Tarjetas de almacenamiento	Micro-SD
Ethernet	No
GPIO	40-Pin
Tamaño	65 x 30 mm
Peso	45 g
Precio inicial	\$ 5

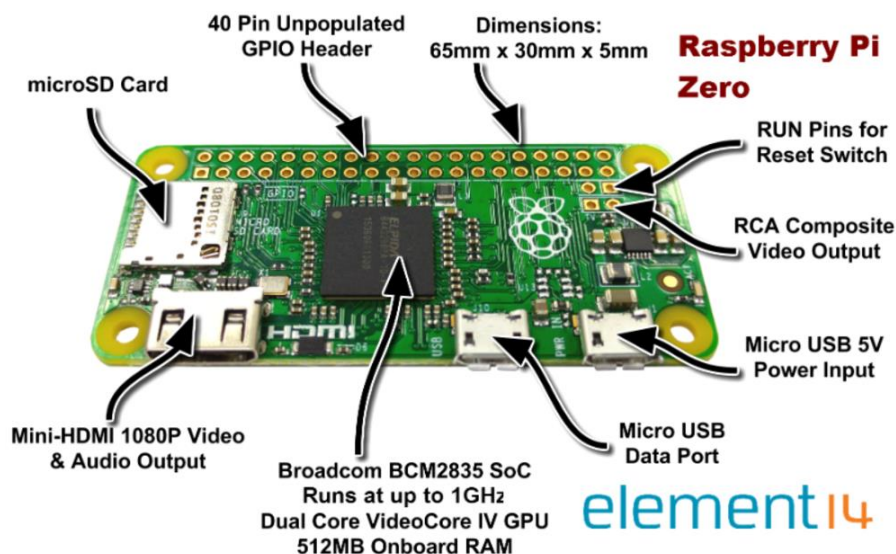


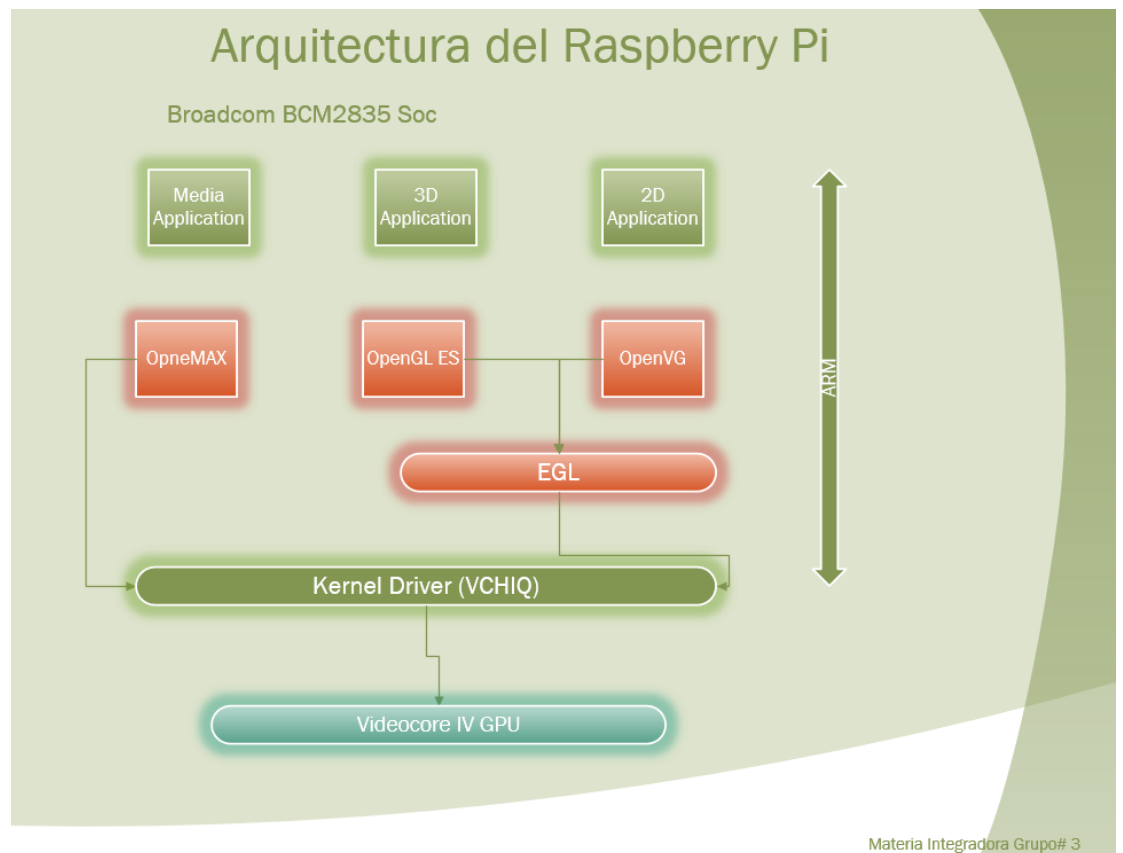
Figura 2.6: Reflexión de un rayo de luz. [21]

## 2.4. Procesador ARM

Este procesador posee una arquitectura RISC (Reduced Instruction Set Computer) que quiere decir ordenador con conjunto reducidos de instrucciones, la simplicidad de estos procesadores hace que sea ideal para las aplicaciones de baja potencia. El Raspberry Pi posee un procesador multimedia BCM2835 system-on-chip (SoC) que quiere decir que la mayoría de sus componentes del sistema se encuentran alojados dentro de ese único componente. [3]

### 2.4.1. Arquitectura ARM

La arquitectura ARM (Ordenador con Conjunto Reducido de Instrucciones) es un conjunto de instrucciones de 32 bits con mayor uso en el mercado. Su simplicidad hace ideal para aplicaciones de baja potencia, además posee un bus de datos de 32 bits. [3]



**Figura 2.7: Arquitectura Del Raspberry.**

## 2.5 Instaladores y Sistemas Operativos para Raspberry Pi.

A continuación, mencionaremos una lista de los instaladores y sistemas operativos que funcionan sobre Raspberry Pi.

### 2.5.1. Instaladores:

- **NOOBS:** Este instalador nos ofrece, en el primer arranque, un menú de instalación de Raspbian, Pidora y algunas opciones con XBCM.
- **BerryBoot Pi:** Nos permitirá instalar los Sistemas Operativos tanto en la tarjeta SD/ micro SD como en una unidad externa que puede ser USB o un disco duro. Además de los Sistema

Operativos que tiene ya por defecto, permite la instalación de cualquier otra distribución remotamente mediante un cliente VNC. Esto se podrá obtener añadiendo una nueva línea de comando en el fichero cmdline.txt donde se indicará la IP local para nuestra Raspberry Pi, mascarará de red y puerta de enlace.

### 2.5.2. Sistemas Operativos:

- **Raspbian:** Una de las distribuciones más antiguas y utilizadas, posee una enorme comunidad de usuarios, miles de paquetes listos para usar desde el primer momento.
- **Snappy Ubuntu Core Pi 2:** Surge gracias a la colaboración entre Canonical y la Fundación Raspberry Pi. Se lo considera el primer Ubuntu para Raspberry Pi y está orientado a Cloud Computing.
- **Pidora:** Es una versión personalizada de Fedora y su estabilidad es una de sus características más importantes.
- **Windows 10 Pi2:** El centro de desarrolladores de Microsoft ha anunciado la compatibilidad de Windows 10 y Raspberry Pi, se debe instalar una versión de Windows 10 que está diseñada para instalar pequeñas aplicaciones que conecten dispositivos, será gratuita pero no es tan completa como la que se puede instalar en la PC.
- **XBian Pi 2:** Esta distribución nos permite conectar la Raspberry al televisor y disfrutar películas, este apoyado sobre XBCM desarrollada para optimizar las funcionalidades de Media Center.
- **Puppy Linux:** Distribución basada en los binarios de Debian Squeeze para ARM y aún está en modelo de prueba. Esta versión incluye todo el conjunto completo de herramientas ocupando tan solo 86MB. Puppy Linux soporta scanners,

impresora y cámara de fotos. Es posible instalar cualquier aplicación que se encuentre en los repositorios de Debian.

## **2.6 Instalador NOOBS.**

Existe una herramienta para el Raspberry Pi, esta es un sistema operativo en muy fácil de usar, conocido como (NOOBS) New Out-Of-Box Software. Se lo puede descargar de manera gratuita en la página [www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads). Para poder realizar la instalación de NOOBS en una tarjeta SD en blanco con una capacidad de no menos de 8 GB para poder instalar el software. Adicionalmente se requiere un lector de tarjetas SD, el software se encuentra en un archivo zip el cual se abre dando doble clic, luego transferir los datos hacia la tarjeta SD, procedemos a expulsar la tarjeta e insertarla en la ranura del SD en el Raspberry Pi.

## **2.7 Característica del Sistema Operativo Raspbian.**

Es un sistema operativo software de distribución libre bajo licencia GNU/Linux que está basado en la distribución Debian que está diseñado para el hardware de la Raspberry Pi. Este sistema viene con más de 35 mil paquetes pre compilado y finalizado su desarrollo en junio de 2012.

Para que Raspbian sea más ligero al momento de descargar solamente incluye de subconjuntos de software, incluye herramientas para navegar por la web, programación en Python. Se podrá instalar el software adicional mediante el gestor de paquetes de la distribución. Raspbian incluye un entorno de escritorio con el nombre de Lightweight X 11 Desktop Entorno (LXDE), que proporciona una interfaz de usuario amigable y muy fácil de utilizar, luego de esto se deberá ingresar el nombre de usuario y la contraseña que por defecto es PI y Raspbian respectivamente.

Lista de paquetes más trascendentes del Sistema Operativo Raspbian.

#### Accesorios

- Referencias de Debian: Es una guía que proporciona una explicación detallada de la distribución Debian Linux y como los programadores pueden aportar a su desarrollo.
- Administrador de archivos: El administrador de archivos predeterminado del escritorio PCManFM, es un gestor de fichero rápido y robusto, el cual ofrece funciones como la navegación con pestañas con un mínimo uso de recursos. Entre sus más importantes características tenemos:
  - Miniaturas de imágenes.
  - Iconos en el escritorio.
  - Multi idioma.
  - Navegador por pestaña.
  - Multiprocesos para cargar directorios extensos rápidamente.
  - Permite arrastrar y soltar.
  - Los ficheros se pueden arrastrar entre pestañas.
  - Soporte para la eliminación de archivos enviándolos a la papelera.
  - GpicView: Le permite visualizar las imágenes como las de una cámara digital.
  - Leafpad: Es un editor de texto simple que nos permite realizar notas rápidas o escribir programas muy simples.
  - LXTerminal: Permite utilizar la línea de comandos de Linux en una ventana sin tener que salir de la interfaz gráfica de usuario.
  - Root Terminal: Nos coloca automáticamente en una sesión de super usuario y así poder realizar tareas de mantenimiento de sistema.
  - Xarchiver: Nos permite extraer archivos comprimidos en Zip.

#### Gráficos

- Xpdf: Es un visor de archivos en formato PDF.



### Internet

- Dillo: Uno de los navegadores web más veloces disponible para Raspberry.
- Midori: Navegador web rápido y ligero, este navegador es similar a Internet Explorer en Windows.
- Midori Private Browsing: Dando clic en este menú me permitirá navegar en modo privado, lo que significa que no se guardaran los datos en el historial del navegador.
- NetSurf Web browser: Similar al navegador Midori, pero mejor en ciertos tipos de páginas web.
- Wpa\_gui: Es una interface gráfica de usuario que permite crackear redes Wireless con (WPA) Wireless Protected Access.

### Programación

- IDLE: Un Entorno de Desarrollo Integrado (IDE) diseñado específicamente para Python.
- Scrath: Es un entorno para enseñar conceptos básicos de programación a los niños.
- Wolfram: Es un acceso directo para cargar el entorno de programación de Wolfram.

### Herramientas de sonido

- QjackCtl: Es una interface gráfica que permite configurar el software de audio.

### Herramientas del sistema

- Task Manager: Esta es una herramienta que nos permite controlar la cantidad de memoria libre y la actual carga de trabajo del procesador en el Raspberry Pi.

#### Preferencias del sistema

- Personalizar apariencia: Es un conjunto de herramientas que permite ajustar el entorno del GUI.
- Configuración de sesión de escritorio: Nos permite cambiar el funcionamiento del sistema cuando el usuario inicia sesión, tales como que programan se cargan automáticamente.
- Administrador de configuración de Openbox: A través de esta herramienta se puede cambiar la apariencia del GUI o alterar el funcionamiento de ciertos aspectos de la interfaz.
- Preferencia de aplicaciones: Una herramienta que me permite cambiar el valor predeterminado que se abren las aplicaciones.

## 2.8 Competitividad en el mercado.

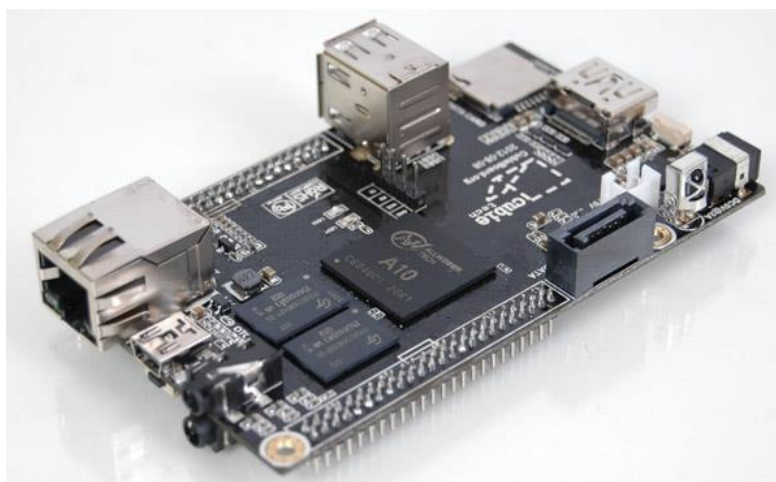
A continuación, mencionaremos algunas alternativas que existen en el mercado, en cuanto a la competitividad que actualmente existe y que son alternativas a la placa Raspberry Pi:

### 2.8.1. Cubeboard

Es una placa que posee un SOC Allwinner A80, que incluye cuatro núcleos ARM Cortex-A15 y otros cuatro ARM Cortex-A7. A continuación mencionaremos sus principales características:

**Tabla 8: Características del CUBIEBOARD.**

<b>CUBIEBOARD</b>	<b>CARACTERÍSTICAS</b>
Procesador	Allwinner A10 a 1000MHz (1 GHz)
Memoria RAM	1 GB DDR3 a 480 MHz
Video	HDMI
Audio	Salida audio por HDMI Salida Jack 3.5 mm Spdif-out
Tarjeta de red	10 / 100 ethernet
USB 2.0	1 mini USB de tipo OTG 2 entradas USB 2.0 de tamaño estándar. Tipo USB HOST
Alimentación	5V / 2ª
Sistemas Operativos	Berryboot Ubuntu 12.04 OpenElec
Tamaño	10X6 cm
Extras	Conexión para disco duro SATA Conector de expansión 2X48 pines Cable DC/USB y cable SATA
Costo (Amazon.com)	\$ 59.99



**Figura 2.8: CUBIEBOARD. [22]**

### 2.8.2. Banana Pi

Un miniordenador dotado de un hardware muy potente, las características que ofrece Banana Pi [7]:

**Tabla 9: Características BANANA PI.**

<b>BANANA PI</b>	<b>CARACTERÍSTICAS</b>
Procesador	ARM7 Dual Core 1 GHz.
Memoria RAM	1 GB DDR3
GPU	ARM MALI400
Ethernet	Gigabit
Conexión	Infrarroja
Audio	Micrófono
Sistemas Operativos	Debian
Tamaño	10X6 cm

Extras	Conexión para disco duro SATA
Costo (Amazon.com)	\$ 47.49



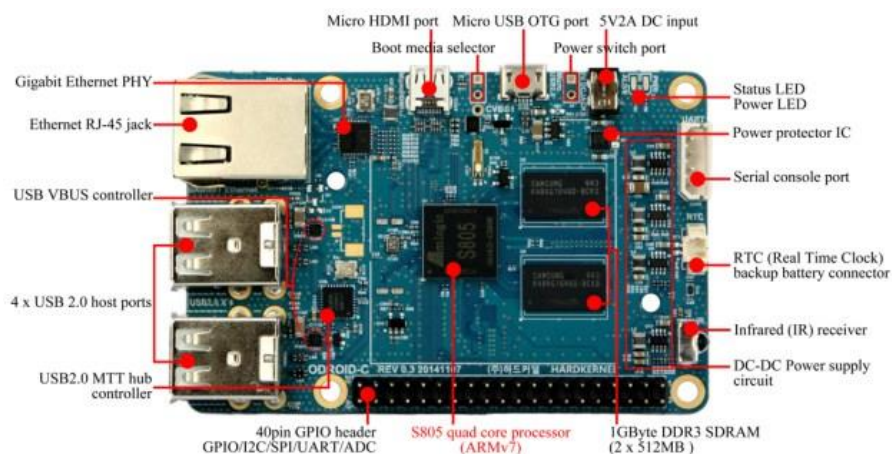
**Figura 2.9: BANANA PI. [8]**

### 2.8.3. ODROID U3

Esta placa es fabricada por la empresa coreana Hard Kernel y presentaremos las características a continuación [8]:

**Tabla 10: Características de ODROID-U3.**

ODROID U3	CARACTERÍSTICAS
Procesador	Samsung Exynos 4412 Prime ARM Cortex-A9 <b>Quad Core 1.7GHz</b>
Memoria RAM	2GB DDR2
GPU	ARM <b>Mali-400</b> Quad Core 440MHz
Ethernet	<ul style="list-style-type: none"> <li>10M/100M Ethernet</li> </ul>
Almacenamiento	Emmc - MICROsd
Sistemas Operativos	Ubuntu Xubuntu 13.10
Conexiones	2 X USB HOST / 1 X micro USB OTG / 1 salida tipo Jack audio
Alimentación	DC5V @ 2A
Costo (Amazon.com)	\$ 49.99



**Figura 2.10: ODROID-U3. [23]**

**Tabla 11: Costo de materiales.**

<b>DISPOSITIVO</b>	<b>COSTO</b>
Raspberry Pi 2 x3	\$180,00
Cable mini USB x6	\$3,50
Router x2	\$25,00
Hub x2	\$24,00
Cinta doble pegatina	\$7,50
Extensión USB x2	\$10,00
Cable Ethernet x6 - 20 cm	\$30,00
Mano de obra	\$108,80
<i>Costo Total</i>	<i>\$388,80</i>

## 2.9 Computación Paralela

La computación paralela es una técnica de programación en la que muchas instrucciones se ejecutan simultáneamente [9]. Gracias a la programación en paralelo se pueden realizar gran cantidad de operaciones simultáneamente.

**Tabla 12: Niveles de computación paralela.**

<b>NIVEL</b>	<b>TÉCNICAS DE IMPLEMENTACIÓN</b>
Paralelismo a nivel de Procesador	<ul style="list-style-type: none"> <li>• Segmentación</li> <li>• División funcional</li> <li>• Procesadores vectoriales</li> </ul>
Paralelismo en Multiprocesadores	<ul style="list-style-type: none"> <li>• Memoria compartida</li> <li>• Memoria distribuida</li> </ul>
Paralelismo en Multicomputadoras	<ul style="list-style-type: none"> <li>• Clúster</li> <li>• Sistemas distribuidos</li> </ul>

## **2.10. Fundamentos Generales de un Clúster**

Debido al gran avance en el campo de la informática, el crecimiento exponencial del Internet, el acelerado incremento del comercio electrónico crea la necesidad en la grandes y pequeñas empresas de garantizar el funcionamiento de su infraestructura tecnológica como: servicios, aplicaciones, seguridad, integridad, confiabilidad y calidad de servicio. Para esto son necesarios los sistemas de redundancias que nos garanticen el funcionamiento del sistema.

Es aquí donde el Clúster nos brindara la solución para mantener el funcionamiento ininterrumpido del sistema y garantizar un excelente servicio al cliente.

### **2.10.1. Concepto**

Hay una gran cantidad de conceptos de clúster, pero en este proyecto la definición de un clúster se entenderá de manera general y en términos muy simples como la unión de un grupo de



equipos conectados entre sí por medio de redes de interconexión, que ejecutan una serie de aplicaciones de manera conjunta y que aparecen como un único recurso ante aplicaciones y clientes.

### **2.10.2. Características**

Existen algunas características que el clúster debe cumplir que serán mencionadas a continuación:

- Para que pueda existir un clúster se debe contar por lo menos con dos nodos capaces de ejecutar las aplicaciones ofertadas por el mismo. No es prescindible que el Clúster sea homogéneo, quiere decir que los nodos pueden ser diferentes.
- Debe existir un medio de comunicación entre los nodos que conforman el Clúster, mediante una red Ethernet dedicada de alta velocidad. Por medio de este enlace los nodos del Clúster podrán intercambiar tareas, actualizaciones y estado del programa.
- El Clúster necesita un software de control especializado, que permitan que los programas sobre la plataforma tomen las ventajas de esta tecnología de Clúster.

### **2.10.3. Clasificación**

Podremos clasificar al clúster de acuerdo a su utilidad:

- Alto rendimiento.
- Alta disponibilidad.
- Balanceo de carga.

**Alto rendimiento**

Con este tipo de clúster se busca resolver problemas de aplicaciones que requieren un nivel de procesamiento, esto se logra repartiéndose la carga de los procesos entre un grupo de nodos individuales y que se encuentran interconectados mediante una red de alta velocidad.

**Alta disponibilidad**

Para las empresas mantener su servicio 24-7 es su mayor prioridad es por eso que brindar una alta disponibilidad mediante una colección de nodos que funcionen en conjunto y que todos realicen la misma función. Es así, si existiera un problema en el nodo principal, los demás asumen las tareas del nodo con problemas.

**Balanceo de carga**

Técnica usada para que los servidores de red compartan la carga de trabajo, existe una jerarquía de nodos maestros y nodos esclavos. Los maestros reciben la petición del servidor y mandan las tareas a los esclavos para que estos las ejecuten, devuelvan el resultado y el servidor responde. Este proceso de dividir la carga de trabajo permite tener un mejor tiempo de acceso a las aplicaciones. Si llegara a existir fallas en uno de los servidores no repercute demasiado ya que la función puede ser suplida por el resto.

**2.11. Implementando Linux como Sistema Operativo**

La elección de Linux como sistema operativo para un sistema de clúster se basó en las siguientes características:

Linux es un sistema operativo desarrollado por cientos de programadores en todo el mundo, pero principalmente por Linus Torvalds. Linux soporta una gran cantidad de archivos, tal como son ext2, ext3, ReiserFS.

## **2.12. Definición de Servidor**

Podemos definir como servidor a un software u ordenador donde funciona el software que proporciona cualquier tipo de información a otros ordenadores o personas llamadas clientes.

### **2.12.1. Tipos de Servidores**

Respecto a la clasificación de los servidores informáticos, podemos separarlos en dos grandes grupos: los dedicados completamente al servicio de los clientes en la red; los no dedicados, aparte de prestar servicios a los clientes puede, a su vez, funcionar como estaciones de trabajo. A continuación, mencionaremos algunos tipos [10]:

- **Servidor de Correo:** Almacenan, envían, reciben y realizan toda clase de operaciones que conciernen con el e-mail de los clientes.
- **Servidor Web:** Almacena documentos HTML, imágenes, videos, texto y en general todo tipo de información. Este servidor se encarga de enviar estas informaciones a los clientes.
- **Servidor VPN:** Una VPN (Red Privada Virtual) está basada en un protocolo que se denomina protocolo de túneles, que quiere decir que cifra los datos que se transmiten desde un lado de la VPN hacia el otro [11]. Cuando nos referimos a “túnel” nos indica que los datos se encontraran

cifrados desde el momento que entran a la VPN hasta que salen de ella, por lo que los datos no se encuentran incomprensible para cualquiera que se encuentre fuera de los extremos de la VPN.

### **2.13. Proyecto Hadoop**

Apache Hadoop es un framework que se encuentra basado en JAVA que soporta aplicaciones distribuidas y que permite a las aplicaciones el trabajar con miles de nodos y petabytes de datos [12]. Hadoop nos permite el procesamiento distribuido de una gran cantidad de datos a través de un conjunto de ordenadores que utilizan un modelo de programación sencilla. Este proyecto está compuesto por los siguientes módulos [13]:

- Hadoop Common: Las utilidades comunes que apoyan los otros módulos de Hadoop.
- Hadoop Distributed File System (HDFS): Un sistema de archivos distribuidos que permite el acceso de alto rendimiento a los datos de la aplicación.
- Hadoop YARM: Un macro para la planificación de tareas y gestión de recursos del clúster.
- Hadoop MapReduce: Es un Sistema que está basado en el hilo para el procesamiento paralelo de grandes conjuntos de datos.

## CAPÍTULO 3

### 3. CONFIGURACIÓN DE CLÚSTERS.

#### 3.1. Diseño de un clúster utilizando plataforma Raspberry Pi

Para este proyecto se ha diseñado dos tipos de clúster sobre la plataforma Raspberry Pi, el primero de ellos formado por tres Raspberry Pi B+ y el segundo, que se encuentra formado por tres Raspberry Pi 2, todos utilizando software libre. Por ende, el coste invertido en software sea nulo y esto condiciona al momento de elegir las aplicaciones a instalar. Pese a que el software es gratuito, los sistemas operativos que usamos están en un nivel muy alto en el mercado actual y no tienen nada que envidiar al software pagado.

El primer clúster está formado por tres Raspberry Pi B+, que trabajan con un procesador Broadcom BCM2835 Soc full HD.

Las especificaciones de este primer clúster que está diseñado bajo la plataforma de Raspberry Pi B+ son las siguientes:

**Tabla 13: Características del primer nodo del clúster con Raspberry PI B+.**

ATRIBUTO	CARACTERÍSTICA
Nombre del equipo	<ul style="list-style-type: none"><li>• Nodo-master</li></ul>
Procesador	<ul style="list-style-type: none"><li>• Broadcom BCM2835 Soc full HD</li></ul>
Memoria Ram	<ul style="list-style-type: none"><li>• 512 MB SDRAM 400 MHz</li></ul>
USB	<ul style="list-style-type: none"><li>• Cuatro puertos USB 2.0</li></ul>
Almacenamiento	<ul style="list-style-type: none"><li>• Tarjeta microSD</li></ul>
Energía	<ul style="list-style-type: none"><li>• 600mA hasta 1. 8ª a 5V</li></ul>
Pines GPIO	<ul style="list-style-type: none"><li>• 40</li></ul>

**Tabla 14: Características del segundo nodo del clúster con Raspberry PI B+.**

<b>ATRIBUTO</b>	<b>CARACTERÍSTICA</b>
Nombre del equipo	<ul style="list-style-type: none"> <li>• Nodo-esclavo1</li> </ul>
Procesador	<ul style="list-style-type: none"> <li>• Broadcom BCM2835 Soc full HD</li> </ul>
Memoria Ram	<ul style="list-style-type: none"> <li>• 512 MB SDRAM 400 MHz</li> </ul>
USB	<ul style="list-style-type: none"> <li>• Cuatro puertos USB 2.0</li> </ul>
Almacenamiento	<ul style="list-style-type: none"> <li>• Tarjeta microSD</li> </ul>
Energía	<ul style="list-style-type: none"> <li>• 600mA hasta 1. 8ª a 5V</li> </ul>
Pines GPIO	<ul style="list-style-type: none"> <li>• 40</li> </ul>

**Tabla 15: Características del tercer nodo del clúster con Raspberry PI B+.**

<b>ATRIBUTO</b>	<b>CARACTERÍSTICA</b>
Nombre del equipo	<ul style="list-style-type: none"> <li>• Nodo-esclavo2</li> </ul>
Procesador	<ul style="list-style-type: none"> <li>• Broadcom BCM2835 Soc full HD</li> </ul>
Memoria Ram	<ul style="list-style-type: none"> <li>• 512 MB SDRAM 400 MHz</li> </ul>
USB	<ul style="list-style-type: none"> <li>• Cuatro puertos USB 2.0</li> </ul>
Almacenamiento	<ul style="list-style-type: none"> <li>• Tarjeta microSD</li> </ul>
Energía	<ul style="list-style-type: none"> <li>• 600mA hasta 1. 8ª a 5V</li> </ul>
Pines GPIO	<ul style="list-style-type: none"> <li>• 40</li> </ul>

Para la parte que corresponde a la conexión se utilizó dos conmutadores, el primero de marca TREDNET y el segundo de marca D-link que nos permitió la interconexión de los Raspberry Pi y que así están tengan acceso a internet. Las especificaciones del equipo las mencionamos a continuación:

**Tabla 16: Características de conmutador TREDnet.**

<b>ATRIBUTO</b>	<b>CARACTERÍSTICA</b>
Marca	<ul style="list-style-type: none"> <li>• TRENDnet</li> </ul>
Modelo	<ul style="list-style-type: none"> <li>• TEW-651BR</li> </ul>
WIFI	<ul style="list-style-type: none"> <li>• Compatible con estándares IEEE 802.11g y 8002.11b</li> </ul>
Puerto LAN	<ul style="list-style-type: none"> <li>• 4 puertos LAN Auto-MDIX a 10/100/1000 Mbps</li> </ul>
Puerto Wan	<ul style="list-style-type: none"> <li>• 1 puerto Wan(Internet) a 10/100/1000 Mbps</li> </ul>
Compatibilidad	<ul style="list-style-type: none"> <li>• Windows, Linux y Mac OS</li> </ul>
Cobertura bajo techo	<ul style="list-style-type: none"> <li>• 100 metros</li> </ul>
Cobertura al aire libre	<ul style="list-style-type: none"> <li>• 300 metros</li> </ul>

**Tabla 17: Características de conmutador D-link.**

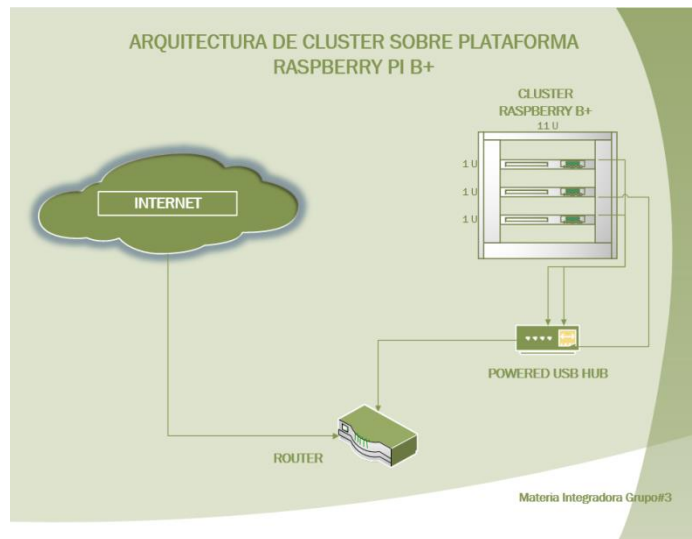
<b>ATRIBUTO</b>	<b>CARACTERÍSTICA</b>
Marca	<ul style="list-style-type: none"> <li>• D-link 610</li> </ul>
Modelo	<ul style="list-style-type: none"> <li>• DIR-610N+</li> </ul>
Velocidad inalámbrica	<ul style="list-style-type: none"> <li>• Hasta 150 Mbps</li> </ul>
Antena interna	<ul style="list-style-type: none"> <li>• Con tecnología Wireless N</li> </ul>
Certificado de la especificación	<ul style="list-style-type: none"> <li>• WPS (Wi-fi Protected Setup)</li> </ul>
Compatible	<ul style="list-style-type: none"> <li>• Usa IPV6</li> </ul>
Puertos Lan	<ul style="list-style-type: none"> <li>• 4 puertos 10/100 Mbps</li> </ul>
Puerto Wan	<ul style="list-style-type: none"> <li>• 1 pu de internet 10/100 Mbps</li> </ul>

### **3.2. Diseño del primer clúster con 3 Raspberry Pi B+**

Este primer Clúster está diseñado sobre la plataforma Raspberry Pi B+, se utilizó tres de estos equipos, además se utilizó herramientas de software libre, tanto para el sistema operativo como para todas las aplicaciones utilizadas. A continuación, se mostrará cómo se encuentra la arquitectura de este primer clúster.



**Figura 3.1. Diseño del primer clúster**



### 3.3. Requerimientos de hardware y software

- Tres Raspberry Pi B+.
- Un USB hub.
- Un router
  - El hardware que se necesitó para implementar el primer clúster de tres nodos fue el siguiente:
- Una conexión de red para lograr la comunicación TCP/IP.
- Acceso a internet.

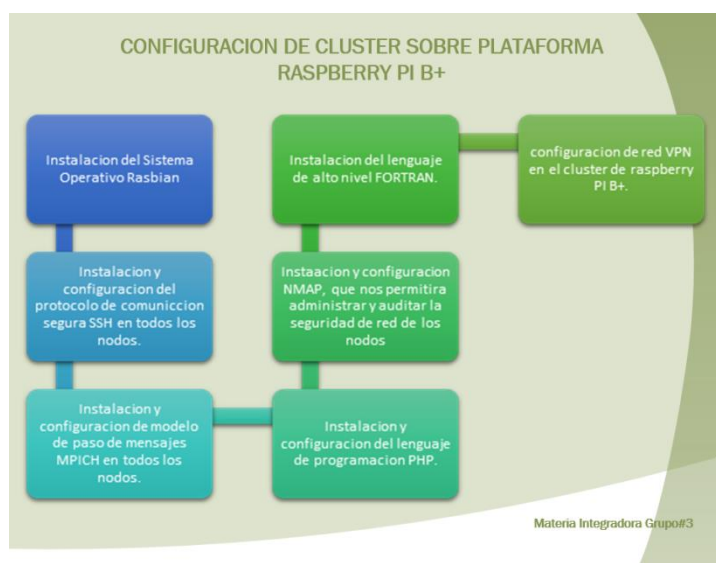
El software a utilizar en el clúster es:

- Distribución de Linux Raspbian como Sistema Operativo.
- Protocolo para la comunicación segura entre los nodos SSH (Secure Shell).
- Estándar del modelo de paso de mensajes MPICH.
- Lenguaje de alto nivel fortran.
- Nmap para administrar y auditar la seguridad de redes de los nodos.
- Lenguaje de programación PHP (Hypertext Preprocessor)

### 3.4. Implementación del primer clúster

La implementación de este clúster de Alta Disponibilidad que proporcionara los servicios de VPN, con un dominio que hemos configurado integradora.espol.ec, se configuraron tres nodos para este clúster, el primero será el nodo maestro, mientras que el segundo y el tercero serán los esclavos 1 y 2 respectivamente.

Para poder migrar los procesos de un nodo a otro, se necesita un soporte TCP/IP de transmisión de datos, se configurará la red local para así poder tener una comunicación ininterrumpida entre los nodos. Mediante el grafico se muestra a continuación se explicará los pasos que se realizaron para la configuración del clúster.



**Figura 3.2. Clúster con tres Raspberry Pi B+**

### 3.5. Introducción a MPI

Message Passing interface (MPI) es un estándar para la implementación de pase de mensaje que nos proporciona una librería para C, C++ o

Fortran que son empleadas en los programas para comunicar datos entre procesos [13].

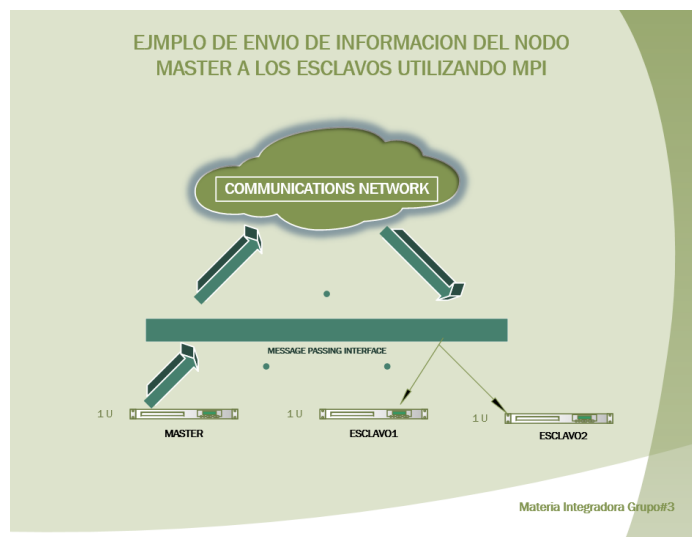
Entre los objetivos importantes mencionamos los siguientes:

- Hacer que la comunicación sea eficiente.
- Permitir que las aplicaciones puedan usarse en ambientes heterogéneos.
- Proveer una interface de comunicación confiable.
- Soportar conexiones de la interface con C y Fortran.

A continuación, mencionaremos las principales características de MPI:

- Iniciar, gestionar y finalizar procesos MPI
- Comunicar de datos entre procesos.
- Realizar operaciones de comunicación entre grupos de procesos. [14]

A continuación, se mostrará un gráfico donde se podrá apreciar cómo se realiza el envío de información a través del nodo master a los esclavos utilizando el paso de mensajería de MPI.



**Figura 3.3. Ejemplo de envío de información del nodo master a los esclavos utilizando MPI.**

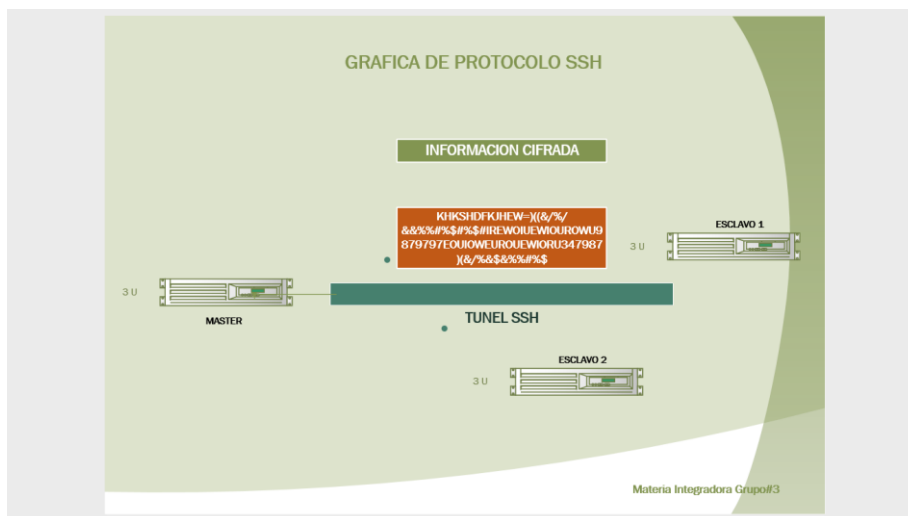
### 3.6. Protocolo que permite conexiones remotas seguras SSH.

SSH es un protocolo que facilita la comunicación y conexión entre dos sistemas. Además, permite que la transmisión de datos se realice de una forma segura, en una arquitectura de cliente/servidor. Este protocolo permite que dos o más nodos de un clúster transfieran información entre ellos. Se debe configurar previamente un puerto para que a través de este SSH pueda escuchar la información que los nodos miembros del clúster envían. Esta cifra los datos que son enviados y esto hace casi imposible que alguien pueda acceder a la información que se está enviando y recibiendo. [13]

A continuación, mencionaremos las principales características del protocolo SSH:

- Iniciar sesiones (login) en servidores remotos.
- Ejecutar comandos remotamente.
- Copiar archivos entre distintos nodos.
- Realizar túneles IP cifrados.

En el siguiente gráfico se mostrará cómo se realiza la comunicación entre los nodos, que reciben y envían información cifrada para para que la transmisión de los datos sea más segura.



**Figura 3.4. Gráfica de protocolo SSH**

### 3.7. Uso de herramienta NMAP (NETWORK MAPPER).

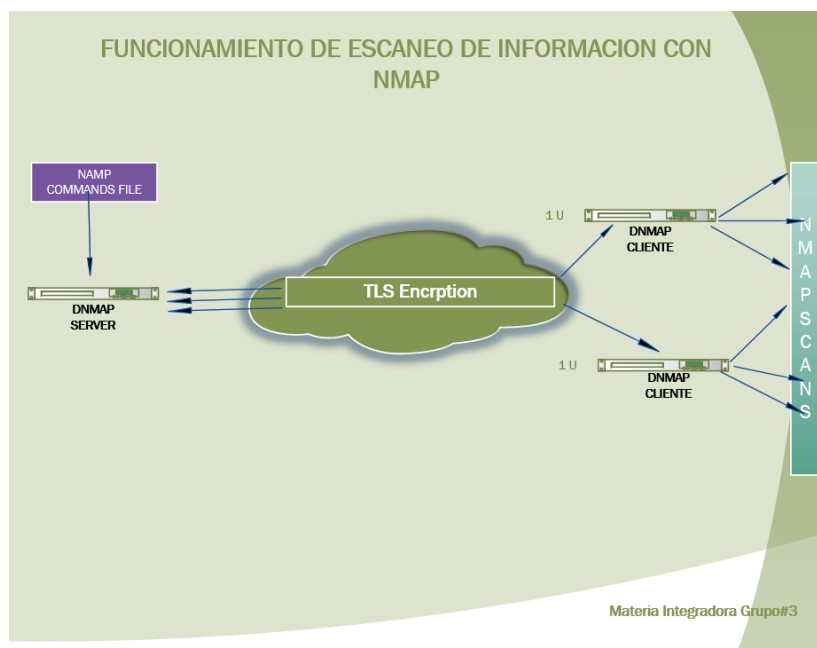
Es una herramienta que permite explorar nuestra red y realizar auditorías de seguridad. Este software libre detecta los nodos que se encuentran online, los puertos que este posee abiertos, los servicios que se encuentran ejecutando en ese momento, su sistema operativo. [14]

Entre los objetivos principales podemos mencionar las siguientes:

- Auditorías de seguridad.
- Pruebas rutinarias de redes.
- Recolectar información para futuros ataques.

A continuación, se muestra un gráfico donde se podrá apreciar el funcionamiento de la herramienta NMAP. Esta herramienta tiene una sintaxis básica.

**Nmap [tipo de escaneo] [opciones]**



**Figura 3.5 Funcionamiento de escaneo con herramienta NMAP**

### 3.8. Instalación y configuración del Sistema Operativo

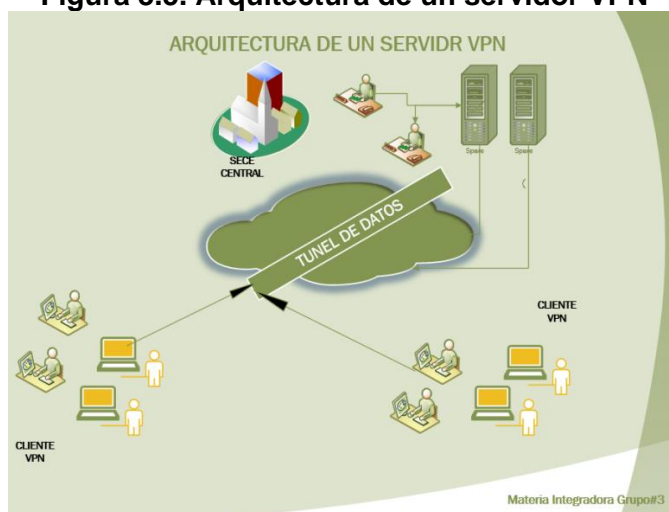
Para la configuración del primer clúster se ha seleccionado el Sistema Operativo Raspbian ya que posee gran cantidad de documentación en internet y no posee costo, no se necesita de ninguna suscripción para poder descargarlo de Internet.

Otra de las razones de elegir Raspbian como sistema operativo del clúster fue por su gran equipo de desarrolladores que proporcionan manuales en línea, foros, tutoriales, entre otros. Raspbian es muy rápido ya que solo ejecuta las versiones básicas de software. El sistema operativo Raspbian puede permanecer una cantidad de tiempo considerable sin requerir ninguna actualización.

### 3.9. Instalación del servicio VPN (VIRTUAL PRIVATE NETWORK)

Un servidor VPN nos brinda la tecnología de red privada virtual que crea una conexión segura a otra red a través de internet. Cuando un dispositivo se conecte a nuestra VPN, es como si estuviese en la misma red y todo el tráfico de datos se realizará de forma segura. [15]

Figura 3.5. Arquitectura de un servidor VPN



### 3.10 Diseño del segundo clúster con 3 Raspberry Pi 2+

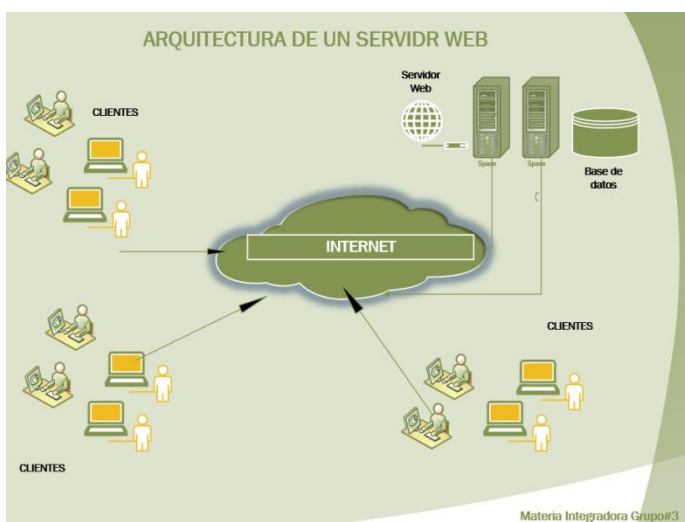


Figura 3.6. Arquitectura de un servidor WEB

<b>PRUEBAS EN CLÚSTER DE RASPBERRY PI</b>			
Prueba de stress	Soporte de un ataque de stress de 10000 peticiones		
Tiempo de respuesta del clúster por el tipo de Raspberry Pi	Tipo B	Tipo 2	
	5 s	2 s	
Criptografía del clúster	Utiliza md5 como método de seguridad al envío de mensajes		
Prueba de archivo en servidor Owncloud	Soporta la carga de un archivo de 5 Gb		

**Tabla 188: Pruebas en clúster.**



## CONCLUSIONES Y RECOMENDACIONES

1. Con el desarrollo de este proyecto se logrará tener flexibilidad en el ensamblaje del clúster, ya que las partes se podrán comprar por separado: monitor, disco duro, entre otros, de acuerdo al gusto y necesidades.
2. El mantenimiento se puede realizar sin complicaciones; los elementos que conforman el clúster se pueden encontrar fácilmente en el mercado. Como el clúster está interconectado con una red, no es necesario poner fuera de servicio todos los nodos para reemplazar un componente o un Raspberry ya que tiene alta disponibilidad.
3. La alta disponibilidad es uno de los grandes atributos de nuestro clúster, si algún componente falla, el nodo que presente inconvenientes quedará fuera de servicio mientras el resto de los nodos podrán atender los requerimientos de los usuarios.
4. Si la demanda de servicios crece en la PYMES, no será necesario eliminar la plataforma existente para incorporar equipos más poderosos, será muy fácil agregar más equipos de Raspberry al clúster para aumentar su capacidad, estos podrán tener mejores prestaciones que los actuales, y se incrementará la capacidad de atención de solicitudes del servicio.
5. El clúster usado con Raspberry permite mantener un servicio estable al menor precio de equipos usados y con un soporte de peticiones muy similar a equipos de clúster de grandes empresas. Con el clúster diseñado bajo la plataforma Raspberry Pi nos servirá para adquirir un nuevo nivel de ejecución e implementación de software libre.
6. Al diseñar un case (chasis) especial para que se adecue al entorno empresarial es necesario tener en cuenta la infraestructura y las especificaciones del cliente para su adecuación y no sufra daño alguno.

7. Se recomienda utilizar micro sd de buena calidad que muestren los GygaBytes completos en su compra, ya que esto puede generar problemas al momento de realizar las réplicas para los demás nodos.

## BIBLIOGRAFÍA

- [1] INEC, (2011, junio). Cens Nacional Económico y Mecanismo de Difusión [online]. Disponible en: [http://www.ecuadorencifras.gob.ec/wp-content/descargas/Presentaciones/resultados\\_generales\\_censo\\_economico.pdf](http://www.ecuadorencifras.gob.ec/wp-content/descargas/Presentaciones/resultados_generales_censo_economico.pdf)
- [2] Revista LIDERES, (2015, diciembre). El sector de las Mipymes esta en pleno crecimiento [online]. Disponible en: <http://www.revistalideres.ec/lideres/sector-mipymes-pleno-crecimiento.html> [1]
- [3] J. Eben Upton and Gareth Halfacree, Raspberry Pi User Guide. Uited States: Eben Upton, 2014.
- [4] K. Ogata, Discrete Time Control Systems, 2<sup>nd</sup> Ed. Englewood Cliffs, NJ: Prentice Hall, 1998.
- [5] Jose Andrade, (2014, julio). El HummingBoard espera competir contra los Raspberry Pi con un CPU desmontable [online]. Disponible en: <http://es.engadget.com/2014/07/03/hummingboard-competencia-raspberrypi/>.
- [6] Rubén Velasco, (2014, abril). Banana Pi, el miniordenado alternativo a Raspberry Pi [online]. Disponible en: <http://www.redeszone.net/2014/04/29/banana-pi-el-miniordenador-alternativo-raspberry-pi/>
- [7] Admin, (2014, marzo). ODROID-U3 [online]. Disponible en: <http://www.uroboros.es/odroid-u3/>
- [8] Jesus Alejandro Moreno Vargas, (2012, noviembre). Computación Paralela [online]. Disponible en: [https://prezi.com/fv\\_uolnmf09x/computacion-paralela/](https://prezi.com/fv_uolnmf09x/computacion-paralela/)
- [8] Manuel Sierra, (2006, julio). QUE ES UN SERVIDOR Y CUALES SON LOS PRINCIPALES TIPOS DE SERVIDORES [online]. Disponible en: [http://aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtp-dhcp&catid=57:herramientas-informaticas&Itemid=179](http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtp-dhcp&catid=57:herramientas-informaticas&Itemid=179)

- [9] CCM, (2015, noviembre). VPN – Redes privadas virtuales [online]. Disponible en: <http://es.ccm.net/contents/258-vpn-redes-privadas-virtuales>
- [10] linbit, (2015, julio). Software Development for High Availability Clúster [online]. Disponible en: <http://drbd.linbit.com/home/what-is-drbd/>
- [11] Andrew Beekhof, (2012, julio). Clúster from Scratch [online]. Disponible en: <http://clústerlabs.org/doc/en-US/Pacemaker/1.1-plugin/html-single/Clústers from Scratch/>
- [12] Francisco Hidrobo y Herbert Hoeger, (2013, Enero). Introduccion a MPI [online]. Disponible en: [http://webdelprofesor.ula.ve/ingenieria/hhoeger/Introduccion\\_MPI.pdf](http://webdelprofesor.ula.ve/ingenieria/hhoeger/Introduccion_MPI.pdf)
- [13] UBUNTU FACIL, (2013, diciembre). CONEXIONES REMOTAS SEGURAS CON SSH [online]. Disponible en: <http://www.ubuntufacil.com/2013/12/conexiones-remotas-seguras-con-ssh/>
- [14] NMAP.ORG, (2013, Enero). Nmap Network Scanning [online]. Disponible en: <https://nmap.org/>
- [15] DiverLandia, (2012, Mayo). Crear servidor VPN en Linux [online]. Disponible en: <http://www.driverlandia.com/crear-servidor-vpn-en-linux/>
- [16] Manuel Sierra, (2013, Febrero). Que es un servidor y cuales son los principales tipos de servidores [online]. Disponible en: [http://aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtp-dhcp&catid=57:herramientas-informaticas&Itemid=179](http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtp-dhcp&catid=57:herramientas-informaticas&Itemid=179)
- [17] EGURIDAD OFENSIVA, (2013, Junio). PENTESTING CON RASPBERRY PI: HARWARE E INSTALACION [online]. Disponible en: <http://www.seguridadofensiva.com/2013/06/pentesting-con-raspberry-pi-hardware-e-instalacion.html>

- [18] CrazyPi, (2013, Junio). Raspberry Pi Model A+ [online]. Disponible en: <https://www.crazypi.com/RASPBERRY-PI-A-PLUS-INDIA>
- [19] Foros de Electronica, (2013, Junio). Raspberry Pi se renueva con su modelo B+ [online]. Disponible en: <http://www.forosdeelectronica.com/f47/raspberry-pi-renueva-modelo-b-119196/>
- [20] Raspberry Shop, (2013, Junio). Ultimas noticias sobre raspberry Pi [online]. Disponible en: <http://www.raspberrystore.com/>
- [21] arstechina (2015, noviembre). Pi Zero: A full raspberry Pi for just \$5 [online]. Disponible en: <http://arstechnica.com/gadgets/2015/11/pi-zero-a-full-raspberry-pi-for-just-4/>
- [22] Robótica, visión artificial, domótica (2013, junio). JdeRobot, tecnología que mola y es útil [online]. Disponible en: <http://blog.jderobot.org/tag/cubieboard/>
- [23] Liliputing (2014, noviembre). Odroid-C1 is a \$35 quad-core [online]. Disponible en: <http://liliputing.com/2014/12/odroid-c1-35-quad-core-single-board-androidlinux-pc.html>

## ANEXOS

### Instalación de imágenes ISO en Raspberry Pi

- Descargar la Imagen Raspaban localizado en:  
<https://www.raspberrypi.org/downloads/raspbian/>
- Quemar la Imagen Raspaban en la tarjeta SD
- Una vez que la imagen se quemó en la tarjeta SD se procede a conectar el Raspberry Pi y arrancar con un teclado, ratón, monitor e Internet adjunta.
- Tras el primer arranque, debería ver la pantalla de configuración del Pi frambuesa (de lo contrario, escriba "sudo raspbi-config")
- Si es necesario, defina las opciones de internacionalización para que coincida con la distribución del teclado países.
- Overclocking del Pi a 800 Mhz
- Opciones avanzadas
- Establezca el nombre de host para PI01
- Dividir la memoria de 16mb para gráficos
- Habilitar SSH

## DESPLIEGUE DEL PLAN DE TRABAJO (DIAGRAMA DE GANTZ)



Figura 3.6. Despliegue de trabajo (Diagrama de Gantz)

## Instalar MPICH

MPICH es un software que permite la comunicación multi-procesamiento entre ordenadores.

Para instalarlo en el Raspberry Pi, primero asegúrese de que tiene una conexión a Internet válida ir al Pi. A continuación, siga estos comandos de terminal:

```
sudo apt-get update
```

```
mkdir mpich2
```

```
cd ~/mpich2
```

```
wget http://www.mpich.org/static/downloads/3.1/mpich-3.1.tar.gz
```

```
tar xzf mpich-3.1.tar.gz
```

```
sudo mkdir /home/rpimpi/
```

```
sudo mkdir /home/rpimpi/mpi-install
```

```
mkdir /home/pi/mpi-build
```

```
cd /home/pi/mpi-build
```

```
sudo apt-get install gfortran
```

```
sudo /home/pi/mpich2/mpich-3.1/configura -prefix=/home/rpimpi/mpi-install
```

```
sudo make
```

```
sudo make install
```

```
nano .bashrc
```

```
PATH=$PATH:/home/rpimpi/mpi-install/bin
```

```
sudo reboot
```

```
mpiexec -n 1 hostname
```

Estos comandos permiten descargar e instalar el MPICH, así como agregarlo como un camino a su archivo de inicio. El último comando se ejecuta como una prueba



para ver si funciona. Si el último comando devuelve el nombre del dispositivo "PI01", entonces se hizo todo correctamente.

#### INSTALANDO MPI4PY

MPICH puede ejecutar programas en C y Fortran. Pero como el Raspberry Pi tiene pre-instalado el entorno de codificación Python, sería más fácil de instalar un Python para que intérprete el MPI. Aquí están los comandos para hacerlo:

```
sudo aptitude install python-dev
```

```
wget https://mpi4py.googlecode.com/files/mpi4py-1.3.1.tar.gz
```

```
tar -zxf mpi4py-1.3.1
```

```
cd mpi4py-1.3.1
```

```
python setup.py build
```

```
python setup.py install
```

```
export PYTHONPATH=/home/pi/mpi4py-1.3.1
```

```
mpiexec -n 5 python demo/helloworld.py
```

#### CONFIGURANDO RASPBIAN

Para cada Raspberry Pi en el clúster, inserte una tarjeta SD de imágenes, conecte todos ellos al mismo router o switch, y luego conectar una fuente de alimentación a cada uno. Se debe configurar el router para mantener una dirección ip reservada y que el raspberry siempre mantenga su conexión aunque se cambie de modem o de internet.

```
Install nmap
```

```
Sudo apt-get update
```

```
Sudo apt-get install nmap
```

```
Get current IP
```

```
Ifconfig
```

```
Scan subnet for Pi's
```

*Sudo nmap -sn 192.168.1.\**

*Run test file*

*Mpiexec -n 1 hostname*

*Make test folder and file*

*Mkdir mpi\_test*

*Cd mpi\_test*

*Nano machinefile*

*192.168.1.2*

*192.168.1.3*

*192.168.1.4*

*192.168.1.5*

*Mpiexec -f machinefile -n 4 hostname*

*Add keys to all pi's*

*PI01*

*Ssh-keygen*

*Cd ~*

*Cd .ssh*

*Cp id\_rsa.pub pi01*

*Ssh pi@192.168.1.3*

*PI02*

*Ssh-keygen*

*Cd .ssh*

*Cp id\_rsa.pub pi02*

*Scp 192.168.1.2:/home/pi/.ssh/pi01 .*

*Cat pi01 >> authorized\_keys*

*Exit*

*PI03 (ssh pi@192.168.1.5)*

*Ssh-keygen*

*Cd .ssh*

*Cp id\_rsa.pub pi03*

*Scp 192.168.1.2:/home/pi/.ssh/pi01 .*

*Cat pi01 >> authorized\_keys*

*Exit*

*PI04*

*Ssh-keygen*

*Cd .ssh*

*Cp id\_rsa.pub pi04*

*Scp 192.168.1.2:/home/pi/.ssh/pi01 .*

*Cat pi01 >> authorized\_keys*

*Exit*

*PI01*

*Scp 192.168.1.3:/home/pi/.ssh/pi02*

*Cat pi02 >> authorized\_keys*

*Run new machinefile*

*Cd ~*

*Nano machinefile*

*192.168.1.2*

*Mpiexec -f machinefile -n 4 hostname*

*Run Python file*

*mpiexec -n 5 python demo/helloworld.py*

*Mpiexec -f machinefile -n 5 python /home/pi/build/mpi4py/demo/md5\_dict\_attack.py*

*Download and unzip file*

*Wget http://www.tinkernut.com/demos/364\_clúster\_comp/python\_test.tar.gz*

*Tar -zxf python\_test.tar.gz*

*Nano python\_test/md5\_attack.py*

*Change hash*

*Copy to all computers*

*Scp -r python\_test 192.168.1.3:/home/pi*

*Mpiexec -f machinefile -n 5 python python\_test/md5\_attack.py*

## INSTALACIÓN DE RED VPN EN CLÚSTER

Una VPN permite desde una red privada conectar a cualquier ubicación geográfica mediante la IP pública, creando un túnel entre el cliente y el servidor, donde los datos cifrados viajan. Siempre y cuando se disponga de los permisos,



autorizaciones y autenticaciones necesarias que permita navegar por el dispositivo como si estuviera en una red local.

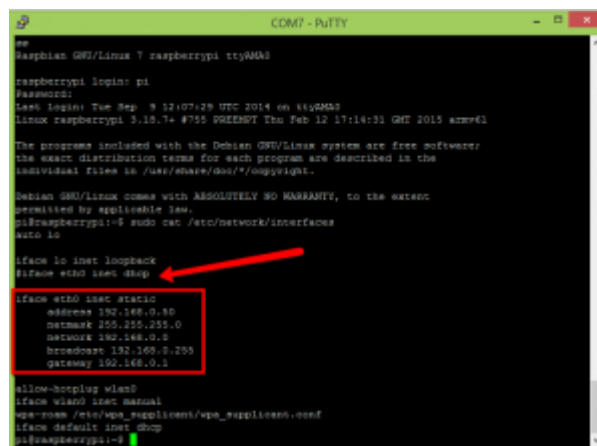
*# Actualizar la RPI*

*sudo apt-get update & sudo apt-get upgrade*

Asegurarse que el servidor (dispositivo) tenga una Ip fija o estática (esto es importante porque cuando un paquete llega al router, este debe ser enviado a la Ip correcta donde está el servidor), esto se puede definir en el mismo router o en el dispositivo editando el siguiente archivo:

*sudo vim.tiny /etc/network/interfaces*

*sudo vim.tiny /etc/network/interfaces*



```

COM7 - PuTTY
ssh
Raspbian GNU/Linux 7 raspberrypi ttyAMA0
raspberrypi login: pi
Password:
Last login: Tue Sep  9 12:07:28 UTC 2014 on ttyAMA0
Linux raspberrypi 3.10.74 #755 PREEMPT Thu Feb 12 17:14:31 GMT 2015 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi:~$ sudo cat /etc/network/interfaces
auto lo
iface lo inet loopback
iface eth0 inet dhcp

iface eth0 inet static
    address 192.168.0.30
    netmask 255.255.255.0
    network 192.168.0.1
    broadcast 192.168.0.255
    gateway 192.168.0.1

allow-hotplug wlan0
iface wlan0 inet manual
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
pi@raspberrypi:~$

```

*# Instalar el protocolo pptp*

*sudo apt-get install pptpd*

*# Editar el archivo:*

*sudo vim.tiny /etc/pptpd.conf*

*# Instalar el protocolo pptp*

*sudo apt-get install pptpd*

*# Editar el archivo:*

*sudo vim.tiny /etc/pptpd.conf*

Agregar la Ip estática asignada anteriormente y el rango de IPs al final del archivo que se le van a asignar a los clientes. Ejemplo:

*localip 192.168.0.50*

*remoteip 192.168.0.234-238,192.168.0.245*

Agregar a este archivo las siguientes descripciones (usar el gateway de la Ip estática a usar):

*ms-dns 192.168.0.1*

*nobsdcomp*

*noipx*

*mtu 1490*

*mru 1490*

Agregar uno o varios nombres de usuarios y las contraseña que deberán tener los clientes que accederán al servidor VPN:

*# Agregarlos usando este formato:*

*NOMBRE\_USUARIO [TAB] \* [TAB] CONTRASEÑA [TAB]\**

*sudo vim.tiny /etc/ppp/chap-secrets*

*# Reiniciar pptpd*

*sudo service pptpd restart*

*# Configurar el tráfico forward en la RPI. Descomentar la línea: net.ipv4.ip\_forward=1*

*sudo nano /etc/sysctl.conf*

*# Aplicar los cambios del forward*

*sudo systemctl -p*

## SERVIDOR WEB EN CLÚSTER

*# Actualizar su Raspberry Pi*

*sudo apt-get update && sudo apt-get upgrade*

*# Instalar Apache*

*sudo apt-get install apache2*

*# Instalar PHP*

*sudo apt-get install php5 libapache2-mod-php5*

*# Reiniciar el servidor de Apache*

*sudo service apache2 restart*

*# Actualizar su Raspberry Pi*

*sudo apt-get update && sudo apt-get upgrade*

*# Instalar Apache*

*sudo apt-get install apache2*

*# Instalar PHP*

*sudo apt-get install php5 libapache2-mod-php5*

*# Reiniciar el servidor de Apache*

*sudo service apache2 restart*

Ahora tienes un Servidor Web corriendo en su Raspberry Pi. Para comprobar el correcto funcionamiento del mismo, tan sólo debes ir a la dirección `/var/www` y ahí encontrará el archivo `index.html` y si quiere ver esto en un navegador de la RPI,

ingrese localhost o 127.0.1.1 en el mismo (ver imagen abajo), o sea, ahí se alojará su nueva página web.

Por otro lado, si su computadora o teléfono celular está conectado a la misma red local de la RPI, con la Ip de la PRI la puede ingresar en cualquier navegador para observar la misma página web, como se muestra en la imagen de abajo:



## It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

### INSTALAR UN SERVIDOR LAMP

LAMP es un acrónimo que representa un conjunto de herramientas que proporcionan las funcionalidades necesarias para construir nuestro propio servidor web. Estas herramientas que forman LAMP son las siguientes:

L de Linux: Por lo tanto nuestro servidor web se usará sobre un sistema operativo Linux.

A de Apache: Por lo tanto el servidor que usaremos es el archiconocido Apache.

M de MySQL/MariaDB: Por lo tanto el servidor web Apache se alimentará de un sistema de base de datos relacional como MySQL o MariaDB.

P de Php, Perl o Phyton: Por lo tanto el servidor web dispondrá de los módulos PHP, Perl o Phyton para poder procesar alguno de estos lenguajes de programación.

Todo el proceso de instalación y configuración del servidor se realizará siendo root. Por lo tanto el primer paso es loguearnos como usuario root. Para ello **en la terminal ejecutamos el siguiente comando:**

*su root*

El primer paso para actualizar el software del sistema operativo donde instalaremos el servidor web, es actualizar los repositorios del sistema. Para ello ejecutaremos el siguiente comando en la terminal:

*apt-get update*



Seguidamente actualizamos los paquetes de nuestro sistema operativo ejecutando el siguiente comando en la terminal:

```
apt-get upgrade
```

Instalar un servidor web apache es muy sencillo. Tan solo tenemos que teclear el siguiente comando en la terminal:

```
apt-get install apache2
```

Una vez teclado presionamos Enter. Justo después empezará el proceso de instalación. Durante el proceso de instalación, recomiendo que analicen los mensajes que aparecen para asegurarnos que no se produce ningún error.

Si durante el proceso de instalación del servidor Apache nos aparece el siguiente mensaje:

```
apache2: Could not determine the server's fully qualified domain name,  
using 127.0.0.1 for ServerName
```

Entonces deberemos editar el fichero `/etc/apache2/httpd.conf`. Para ello tenemos que teclear el siguiente comando en la terminal:

```
nano /etc/apache2/httpd.conf
```

Una vez abierto el fichero `httpd.conf` con el editor de textos nano, tenemos que ir al final del archivo de configuración y añadir el siguiente texto:

```
ServerName localhost
```

Una vez introducida esta línea guardamos los cambios y cerramos el editor de texto. Ahora al reiniciar el servidor apache no debería aparecer el error ya que ahora hemos definido que el nombre de nuestro servidor sea localhost.

En estos momentos el proceso de instalación del servidor Web ha finalizado.

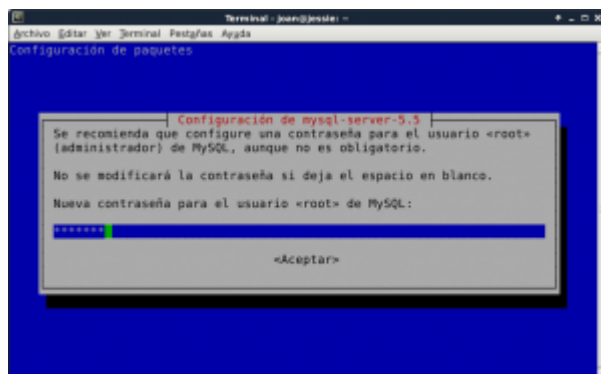
Una vez instalado el servidor web Apache, instalaremos las librerías de soporte PHP conjuntamente con sus dependencias. Para ello ejecutamos el siguiente comando en la terminal:

```
apt-get install php5 libapache2-mod-php5 php5-mcrypt
```

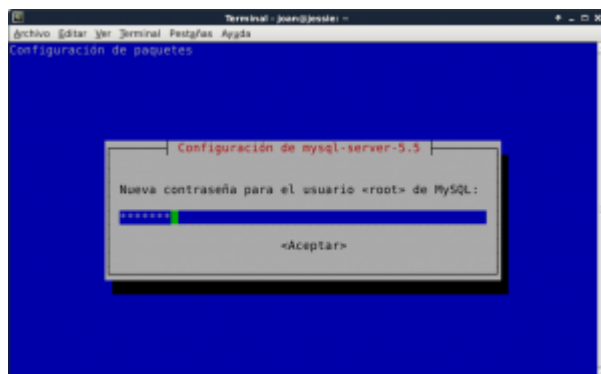
El siguiente paso es instalar el servidor de base de datos MySQL. Para ello ejecutaremos el siguiente comando en la terminal:

```
apt-get install mysql-server php5-mysql
```

Durante la instalación del servidor de la base de datos, tal y como se puede ver en la captura de pantalla, se nos pedirá introducir la contraseña del usuario root que administrará la base de datos.



Tal y como se puede ver en la captura de pantalla, introducimos la contraseña y presionamos Enter. Justo después de presionar Enter, tal y como se puede ver en la captura de pantalla, se nos volverá a pedir que repitamos la contraseña que acabamos de introducir:

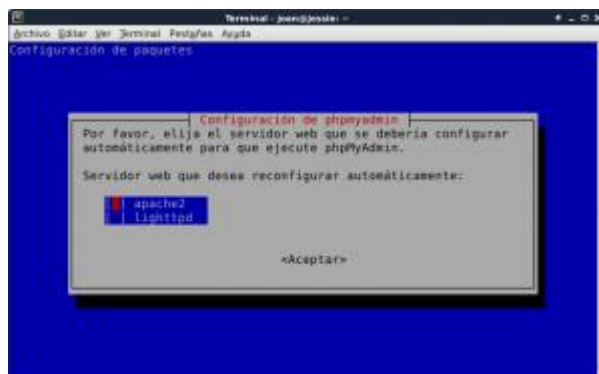


Introducimos la contraseña de nuevo y presionamos Enter. Ahora tan solo tenemos que esperar unos segundos a que concluya el proceso de instalación de MySQL.

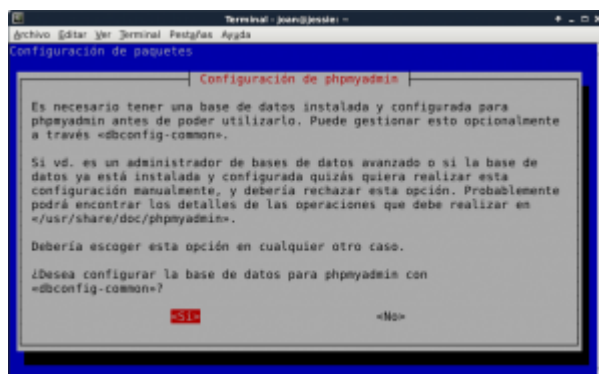
Phpmyadmin es un administrador gráfico web para bases de datos MySQL. Por lo tanto Phpmyadmin nos servirá para poder administrar de una forma gráfica y más sencilla nuestras bases de datos. Para instalar Phpmyadmin tenemos que ejecutar el siguiente comando en la terminal:

```
apt-get install phpmyadmin
```

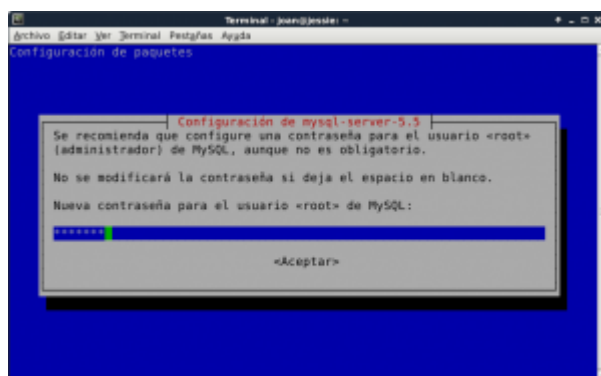
Durante la instalación de Phpmyadmin se nos preguntará el servidor web en el que queremos ejecutar Phpmyadmin. Tal y como se puede ver en la captura de pantalla, seleccionamos el servidor apache2 que acabamos de instalar y presionamos Enter.



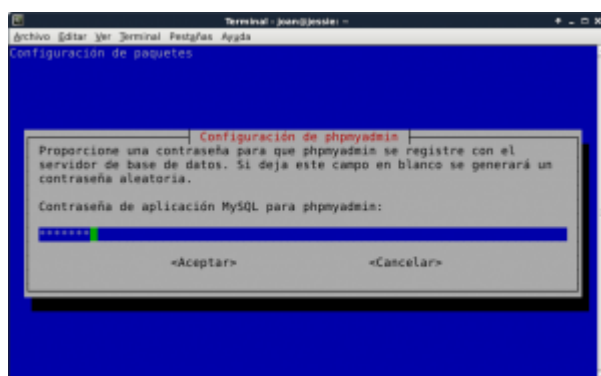
Seguidamente nos aparecerá un mensaje en el que se nos advierte que es necesario disponer de una base de datos instalada y configurada para poder utilizar phpmyadmin. Se nos pregunta si queremos que la creación y configuración de esta base de datos se haga de forma automática. Nosotros, tal y como se puede ver en la captura de pantalla, seleccionaremos la opción Sí y presionaremos Enter.



Después de presionar Enter continuará el proceso de instalación. En breves momentos aparecerá otra ventana en la que se nos pedirá que introduzcamos la contraseña de administrador root de Mysql para que phpmyadmin pueda acceder al servidor de base de datos Mysql y crear la base de datos. Tal y como se puede ver en la captura de pantalla, introducimos la contraseña que definimos previamente y presionamos Enter:



Después de presionar Enter continuará el proceso de instalación. En breves momentos aparecerá otra ventana en la que nos pedirá que introduzcamos la contraseña que queremos usar para loguearnos a phpmyadmin. Tal y como se puede ver en la captura de pantalla introducimos la contraseña y presionamos la tecla Enter.



Al presionar Enter nos aparecerá otra pantalla en la que se nos pedirá que reconfirmemos la contraseña que acabamos de introducir. Por lo tanto, volvemos a teclear la contraseña y presionamos Enter. En estos momentos el proceso ha finalizado.

Finalmente, tan solo nos falta incluir phpmyadmin dentro de la configuración de apache. Para ello ejecutamos el siguiente comando en la terminal:

```
nano /etc/apache2/apache2.conf
```

Una vez abierto el fichero de configuración de Apache, nos vamos al final e introducimos el siguiente texto:

```
# phpMyAdmin Configuración
```

```
Include /etc/phpmyadmin/apache.conf
```

Una vez introducido el texto guardamos los cambios y cerramos el fichero. Finalmente reiniciamos el servidor Apache para los cambios surjan efecto introduciendo el siguiente comando en la terminal:

`service apache2 restart`

Para comprobar que el servidor web está funcionando, tan solo tenemos que abrir el navegador web y teclear la dirección de nuestro dominio no-ip fuera de nuestra red local.

`http://192.168.10.1`

Una vez tecleada la dirección, si todo funciona adecuadamente obtendréis un resultado parecido al siguiente:



Para comprobar que el funcionamiento de php es correcto tenemos que crear un pequeño script. Para crearlo ejecutamos el siguiente comando en la terminal:

`nano /var/www/html/info.php`

Una vez abierto el editor de texto nano tecleamos el siguiente código:

```
<?php
phpinfo();
?>
```

Una vez tecleado el código, guardamos los cambios y cerramos el archivo.

Una vez generado el script ya podemos realizar la comprobación. Para realizar la comprobación, tan solo tenemos que abrir el navegador web con un dispositivo que esté fuera de nuestra red local y teclear la dirección de nuestro dominio no-ip seguida de una contrabarra y el nombre del script que acabamos de generar. Por lo tanto en mi caso tengo que teclear la siguiente dirección:

`http://192.168.10.1/info.php`

Una vez tecleada la dirección, si todo funciona adecuadamente, obtendréis un resultado parecido al siguiente:

PHP Version 5.3.18-1ubuntu2	
Package	php5.3.18-1ubuntu2
Architecture	amd64
Version	5.3.18-1ubuntu2
PHP Module Support	all
Configuration File	/etc/php5/apache2
Configuration File	/etc/php5/apache2/php.ini
Configuration File	/etc/php5/apache2/conf.d
Configuration File	/etc/php5/apache2/conf.d/05-opcache.ini
Configuration File	/etc/php5/apache2/conf.d/10-pdo.ini
Configuration File	/etc/php5/apache2/conf.d/20-curl.ini
Configuration File	/etc/php5/apache2/conf.d/20-gd.ini
Configuration File	/etc/php5/apache2/conf.d/20-imagick.ini
Configuration File	/etc/php5/apache2/conf.d/20-intl.ini
Configuration File	/etc/php5/apache2/conf.d/20-json.ini
Configuration File	/etc/php5/apache2/conf.d/20-mcrypt.ini
Configuration File	/etc/php5/apache2/conf.d/20-mysql.ini
Configuration File	/etc/php5/apache2/conf.d/20-pdo_mysql.ini
Configuration File	/etc/php5/apache2/conf.d/20-readline.ini
Version	20131106
Version	20131226
Version	220131226
Architecture	API20131226,NTS
Architecture	API20131226,NTS
Architecture	no
Architecture	disabled
Architecture	disabled

Para comprobar que MySQL y Phpmyadmin están funcionando, tan solo tenemos que abrir el navegador web con un dispositivo que esté fuera de nuestra red local y teclear la dirección de nuestro dominio no-ip seguida de una contrabarra y el nombre phpmyadmin. Por lo tanto en mi caso tengo que teclear la siguiente dirección:

*<http://192.168.10.1/phpmyadmin>*

Nota: Si la prueba se realiza dentro de nuestra red local, se debe sustituir la dirección del dominio no-ip por la dirección ip estática de nuestro servidor web.

Una vez tecleada la dirección, si todo funciona adecuadamente obtendréis un resultado parecido al siguiente:



INSTALAR OWNCLOUD AL CLÚSTER

Owncloud es una nube personal como lo puede ser por ejemplo Dropbox, OneDrive o Google Drive.

La gran diferencia entre los servicios que acabo de citar y Owncloud es que este último es una aplicación de software libre, y por lo tanto tiene la ventaja que nos dará el control total de nuestros datos en la nube. De este modo evitaremos los problemas de privacidad que pueden generar servicios de terceros.

Las opciones que nos ofrece owncloud son mucho más grandes de lo que a priori uno puede pensar. Parte de las funcionalidades que nos dará Owncloud son las siguientes:

- Almacenamiento de archivos cifrados en la nube.
- Compartición de archivos/información ubicada en la nube.
- Servidor de archivos WebDAV.
- Galería de imágenes.
- Reproductor de archivos de vídeo y sonido, etc.
- Editor de texto en línea que además permite la redacción de documentos colaborativos.
- Visor de archivos pdf, odt, archivos de imagen, etc.
- Sincronización de nuestra información en la totalidad de equipos informáticos que tengamos.
- Calendario y agenda de contactos mediante los protocolos CalDAV y CardDAV. De este modo será sumamente sencillo sincronizar nuestro calendario y agenda en nuestros dispositivos móviles y resto de equipos.

Una vez instalada toda la infraestructura para el funcionamiento de owncloud, ahora ya lo podemos instalar.

Para que owncloud funcione adecuadamente tenemos que asegurar que una serie de paquetes estén instalados. Para ello ejecutamos los siguientes comandos en la terminal:

```
apt-get install php5-gd php5-json php5-curl
```

```
apt-get install php5-intl php5-imagick
```

El siguiente paso es descargar y descomprimir el Software Owncloud. Para ello creamos y accedemos a la carpeta donde descargaremos owncloud ejecutando los siguientes comandos en la terminal:

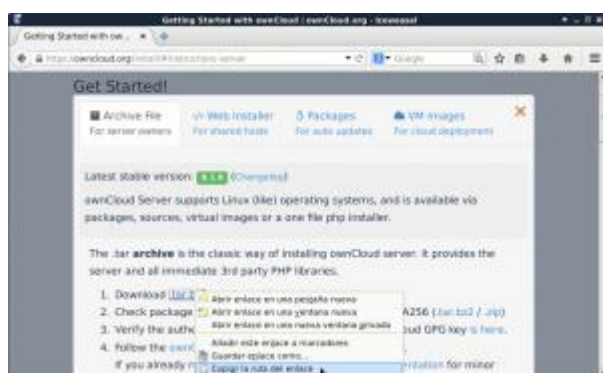
```
mkdir /home/owncloud
```

```
cd /home/owncloud
```

Para descargar Owncloud accedemos a la siguiente página web:

```
https://owncloud.org/install/#instructions-server
```

Una vez dentro de la página web, tal y como se puede ver en la captura de pantalla, posicionamos el puntero del mouse encima del link `.tar.bz2`, presionamos el botón derecho del mouse y cuando se despliegue el menú seleccionamos la opción Copiar dirección de enlace y presionamos el botón izquierdo del mouse.



Seguidamente en la terminal, y dentro de la ubicación `/home/owncloud`, tecleamos el comando `wget` y seguidamente pegamos la dirección de descarga de owncloud. Por lo tanto en mi caso el comando para descargar owncloud es el siguiente:

```
wget https://download.owncloud.org/community/owncloud-8.0.3.tar.bz2
```

Una vez descargado Owncloud lo descomprimos mediante el siguiente comando:

```
tar -xvf owncloud-8.0.3.tar.bz2
```

Una vez descargado y descomprimido Owncloud, ahora hay que mover la totalidad de archivos descomprimidos a nuestro servidor web. Para ello, dentro de la ubicación `/home/owncloud`, hay que ejecutar el siguiente comando:

```
mv owncloud /var/www/html/
```

La base de datos para Owncloud la crearemos con phpmyadmin. Por lo tanto el primer paso que tenemos que realizar es acceder a phpmyadmin. Para acceder a phpmyadmin lo haremos remotamente a través de un equipo que esté en la misma red local que nuestro servidor web. Por lo tanto en un ordenador diferente al que



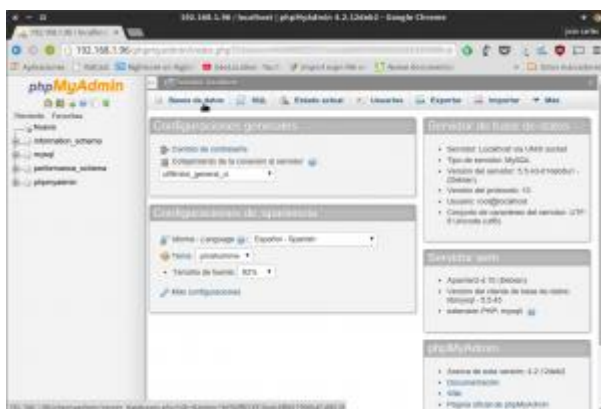
tenemos instalado el servidor web introducimos la siguiente dirección en el navegador web:

*http://192.168.10.1/phpmyadmin*

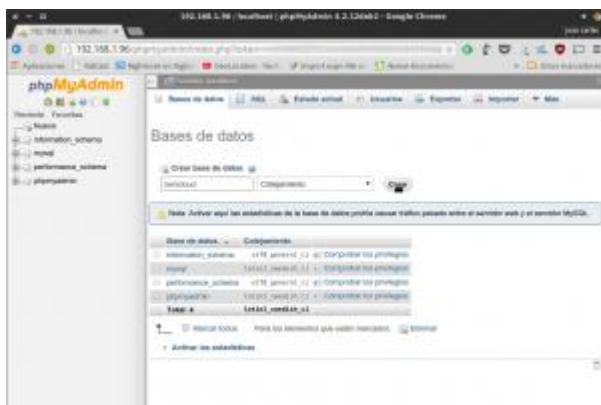
Una vez dentro de la pantalla de login, tal y como se puede ver en la captura de pantalla, introducimos el usuario y la contraseña de phpmyadmin y presionamos le botón Continuar:



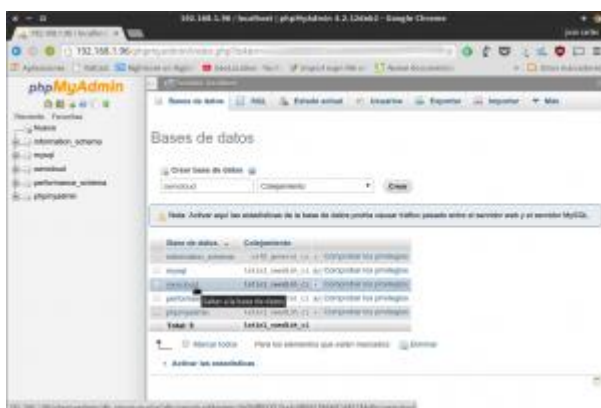
Seguidamente, tal y como se puede ver en la captura de pantalla, presionamos el botón Bases de datos



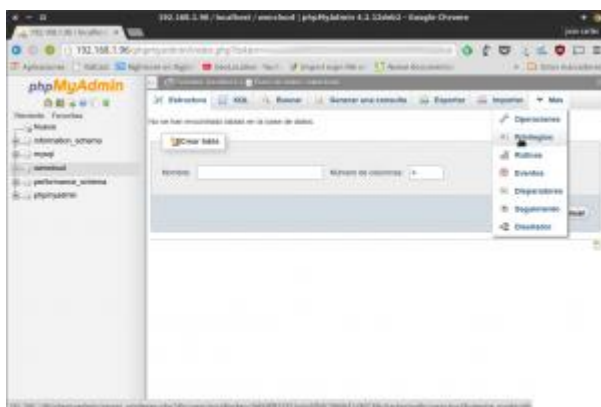
El siguiente paso, tal y como se puede ver en la captura de pantalla, es poner un nombre a la base de datos. Después de poner el nombre, que en mi caso es owncloud, hay que presionar encima del botón Crear.



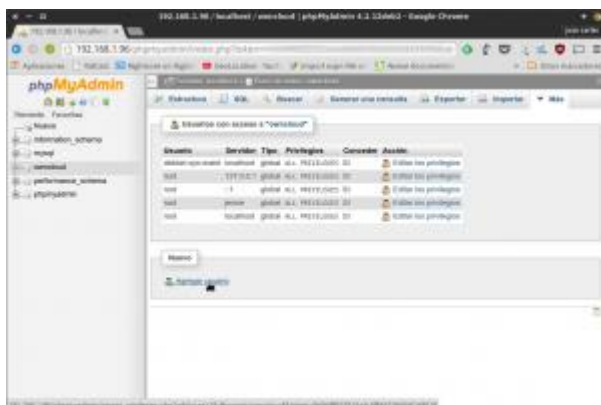
Ahora tenemos que acceder a la base de datos que acabamos de crear. Para ello, tal y como se puede ver en la captura de pantalla, clicamos encima del nombre de nuestra base de datos:



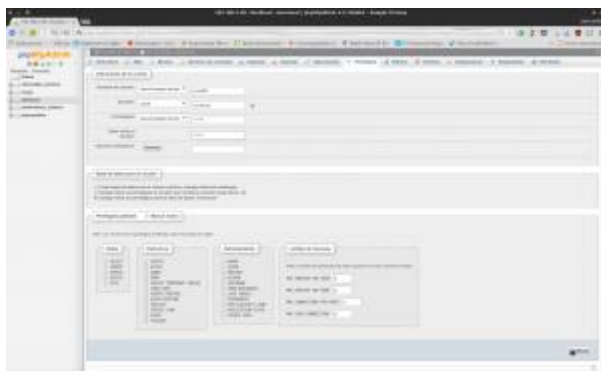
Una vez dentro de nuestra base de datos, tal y como se puede ver en la captura de pantalla, clicamos encima de la pestaña Privilegios:



Seguidamente creamos un usuario. Para ello, tal y como se puede ver en la captura de pantalla, lo primero que tenemos que hacer es clicar encima de Agregar usuario:



Después de clicar encima de la pestaña agregar usuario, tal y como se puede ver en la captura de pantalla, tenemos que seleccionar el nombre del usuario, el tipo de servidor y la contraseña del usuario.



En el campo Nombre de usuario pueden elegir el nombre de usuario que deseen.

En el campo Servidor, tienen que indicar que se trata de un servidor local, para ello en el desplegable Servidor hay que seleccionar la opción Local.

En el campo Contraseña hay que indicar la contraseña de usuario que quieran mientras que en el campo Debe volver a escribir tendremos que repetir de nuevo la contraseña de usuario.

En el apartado Base de datos para el usuario, tenéis que asegurar que la opción Otorgar todos los privilegios para la base de datos "nombre de vuestra base" este marcada

Finalmente una vez rellenados todos los datos tan solo hay que presionar encima del botón Continuar.

Para que el funcionamiento de owncloud sea correcto tenemos que otorgar los permisos adecuados a las carpetas donde ubicamos e instalamos owncloud.

Para ello en el servidor web cambiamos los permisos de forma recursiva de la carpeta `/var/www/html/owncloud` ejecutando el siguiente comando en la terminal.

```
chmod 0755 /var/www/html/owncloud -R
```

Seguidamente asignamos un grupo y un usuario de forma recursiva a la carpeta donde figuran la totalidad de archivos de owncloud. Par ello ejecutamos el siguiente comando en la terminal:

```
chown -R www-data:www-data /var/www/html/owncloud/
```

Ejecutando este comando la totalidad de archivos de instalación de owncloud perteneceran al usuario `www-data` y al grupo `www-data`.

Por motivos de seguridad quiero que la ubicación del almacenamiento de los archivos de los usuarios de Owncloud sea fuera de la ruta de instalación de Owncloud. Para ello crearemos la carpeta `datosowncloud` en nuestra ubicación `home` que es la que almacenará los datos de los distintos usuarios de la nube. Para crear la carpeta `datosowncloud` ejecutamos el siguiente comando en la terminal:

```
mkdir /home/datosowncloud
```

Seguidamente otorgaremos los permisos necesarios a la carpeta `datosowncloud` ejecutando el siguiente comando en la terminal:

```
chmod 0755 /home/datosowncloud -R
```

Finalmente asignamos un grupo y un usuario de forma recursiva a la carpeta que almacenará los datos de los clientes de nuestra nube owncloud. Para ello ejecutamos el siguiente comando en la terminal:

```
chown -R www-data:www-data /home/datosowncloud/
```

En estos momentos ya podemos abrir el navegador y probar si nuestro servidor owncloud funciona.

Ha llegado la hora de intentar iniciar owncloud por primera vez. Para ello en un equipo que esté conectado en la misma red local que el servidor, abrimos el navegador y tecleamos la ip del servidor seguida de una barra y la palabra `owncloud`:

```
http://192.168.10.1/owncloud
```

Al presionar `Enter` es probable que obtengan el error que se muestra en la siguiente captura de pantalla:



Para solucionar el error “PHP está configurado para transmitir datos raw”, tal y como nos indica el texto del error, tecleamos el siguiente comando en la terminal del servidor donde se halla instalado owncloud:

```
nano /etc/php5/apache2/php.ini
```

Una vez abierto el editor de textos nano tenemos que localizar la siguiente línea:

```
;always_populate_raw_post_data a -1
```

Una vez localizada la cambiamos por esta:

```
always_populate_raw_post_data a -1
```

Guardamos los cambios, cerramos el archivo y seguidamente tecleamos el siguiente comando en la terminal para reiniciar nuestro servidor web:

```
service apache2 restart
```

Ahora si intentamos acceder de nuevo al servidor owncloud tenemos que obtener el siguiente resultado:



Tal y como se puede ver en la captura de pantalla, en el apartado Crear una cuenta de administrador hay que introducir el nombre de usuario y contraseña del que será

el administrador de la nube. Obviamente podéis seleccionar el nombre de usuario y contraseña que queráis.

Seguidamente en el apartado Directorio de datos especificamos la ubicación de almacenamiento que queramos que tenga la información que subimos/almacenamos a owncloud. Puede ser un disco duro externo incluso o nuestro mismo disco duro. En mi caso, tal y como hemos visto en apartados anteriores elegiré la ubicación /home/datosowncloud. Por cuestiones de seguridad no es recomendable usar la ubicación de almacenamiento predeterminada.

Finalmente, en el apartado Configurar Base de Datos introducimos el nombre del usuario de base de datos que hemos creado antes y ponemos su contraseña. Seguidamente ponemos el nombre de la base de datos y el tipo de servidor que en nuestro caso es Localhost.

Finalmente tan solo falta presionar el botón Completar la instalación para finalizar el proceso de instalación y obtener el siguiente resultado:



En estos momentos ya podemos acceder al servidor owncloud desde nuestra red local. Ahora intentaremos acceder a él conectándonos desde el exterior. Por lo tanto en mi caso abro el navegador de mi teléfono y visito la siguiente URL:

*<http://192.168.10.1/owncloud>*

En el momento de acceder lo más probable es que obtengan el error que se muestra en la captura de pantalla:

Servidor desde un dominio inseguro



Si el contenido ha sido añadido correctamente guardamos los cambios y cerramos el fichero. Ahora si intentamos acceder de nuevo al servidor veremos que ya no se produce el error:



Por lo tanto a estas alturas tenemos configurado el servidor owncloud para acceder desde nuestra red local y desde el exterior de nuestra red local.

En estos momentos Owncloud es funcional. No obstante si os importa la seguridad y sacar el máximo partido al servidor es indispensable implementar los siguientes pasos adicionales:

Hacer que apache utilice el fichero .htaccess

Las instrucciones de este apartado tan solo hay que aplicarlas en el caso que en uno de los apartados de la instalación hayan visto un mensaje que decía lo siguiente:

*“Su directorio de datos y sus archivos probablemente sean accesibles a través de internet ya que su archivo .htaccess no funciona”*

Para solucionar este problema tenemos que configurar Apache para que no ignore las instrucciones contenidas en el archivo .htaccess. Para ello en el servidor Owncloud tecleamos el siguiente comando en la terminal:

```
nano /etc/apache2/sites-enabled/000-default.conf
```

Cuando se abra el editor de texto nano, hay que localizar la siguiente línea:

```
DocumentRoot /var/www/html
```

Justo por debajo de esta línea hay que añadir el siguiente código:

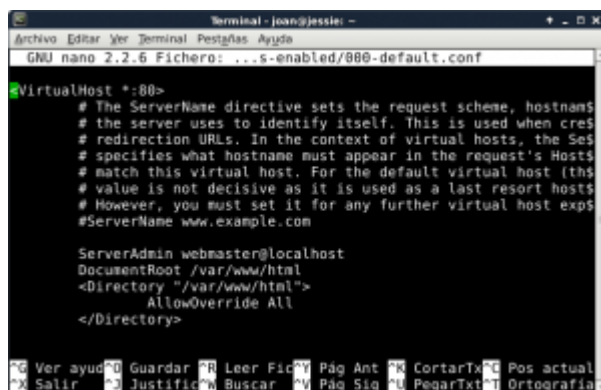
```
<Directory "/var/www/html">
```



```
AllowOverride All
```

```
</Directory>
```

Una vez introducidos los cambios el fichero 000-default.conf tendrá un aspecto parecido al siguiente:



```

GNU nano 2.2.6 Fichero: ...s-enabled/000-default.conf

VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname
# the server uses to identify itself. This is used when cre
# redirection URLs. In the context of virtual hosts, the Se
# specifies what hostname must appear in the request's Host
# match this virtual host. For the default virtual host (th
# value is not decisive as it is used as a last resort host
# However, you must set it for any further virtual host exp
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
<Directory "/var/www/html">
    AllowOverride All
</Directory>

```

El siguiente paso es activar los módulos rewrite y headers. Para activar estos 2 módulos hay que ejecutar los siguientes comandos en la terminal del servidor Owncloud:

```
a2enmod rewrite
```

```
a2enmod headers
```

Una vez activados los módulos tan solo tenemos que reiniciar el servidor apache introduciendo el siguiente comando en la terminal:

```
service apache2 restart
```

Con la configuración estándar del servidor no podremos subir ficheros de más de 2 MB. Para modificar este aspecto tenemos que modificar la configuración de PHP ejecutando el siguiente comando en la terminal:

```
nano /etc/php5/apache2/php.ini
```

Una vez abierto el editor de textos nano localizamos la siguientes líneas:

```
upload_max_filesize = 2M
```

```
post_max_size = 8M
```

Una vez localizadas modificamos sus valores por los siguientes:

```
upload_max_filesize = 2048M
```

```
post_max_size = 2500M
```

Una vez realizados los cambios tan solo hay que guardarlos y cerrar el fichero.

Seguidamente tenemos que configurar el archivo `.htaccess`. Para ello tecleamos el siguiente comando en la terminal:

```
nano /var/www/html/owncloud/.htaccess
```

Una vez abierto el editor textos nano localizamos las siguientes líneas:

```
php_value upload_max_filesize 513M
```

```
php_value post_max_size 513
```

Una vez localizadas modificamos sus valores por los siguientes:

```
php_value upload_max_filesize 2048M
```

```
php_value post_max_size 2500
```

Una vez realizados los cambios tan solo hay que guardarlos y cerrar el archivo. Seguidamente tecleamos el siguiente comando en la terminal para reiniciar nuestro servidor web:

```
service apache2 restart
```

Después de realizar estos pasos seremos capaces de subir archivos de 2 GB en nuestra nube personal Owncloud.

La dirección Owncloud es del tipo http Por lo tanto el tráfico generado entre cliente y servidor no está cifrado. Si queremos añadir una capa de seguridad tan solo tenemos que seguir los siguientes pasos. Primeramente tenemos que asegurar que el paquete openssl está instalado ejecutando el siguiente comando en la terminal:

```
apt-get install openssl
```

El siguiente paso es activar el módulo ssl de apache. Para ello tecleamos el siguiente comando en la terminal:

```
a2enmod ssl
```

A continuación editaremos el fichero de configuración de openssl para alargar la caducidad del certificado ssl que vamos a generar, y para cambiar la ubicación donde se guardará el certificado que generaremos. Para ello ejecutamos el siguiente comando en la terminal:

```
nano /etc/ssl/openssl.cnf
```

Una vez abierto el editor de textos nano localizamos las siguientes líneas:

```
dir = ./demoCA
```

```
default_days = 365
```

Una vez localizadas cambiamos sus valores por los siguientes:

```
dir = /root/SSLCertAuth
```

```
default_days = 3650
```

Una vez aplicados los cambios los guardamos y cerramos el fichero. En estos momentos cuando creamos el certificado se almacenará en la carpeta `/root/SSLCertAuth` y tendrá una vigencia de 3650 días (10 años).

Ahora creamos y configuramos los directorios donde vamos a crear los certificados. Para ello ejecutamos el siguiente comando en la terminal:

```
mkdir /root/SSLCertAuth
```

El siguiente paso es hacer que únicamente el usuario propietario, que en mi caso es `root`, tenga permisos de lectura, escritura y ejecución sobre la carpeta `/root/SSLCertAuth`. Para ello ejecutamos el siguiente comando en la terminal:

```
chmod 700 /root/SSLCertAuth
```

Accedemos dentro de la carpeta `/root/SSLCertAuth` ejecutando el siguiente comando en la terminal:

```
cd /root/SSLCertAuth
```

Seguidamente tenemos que crear una serie de carpetas que son necesarias para que `openssl` pueda generar nuestras claves y certificados. Para ello ejecutamos el siguiente comando en la terminal:

```
mkdir certs private newcerts
```

Una vez creadas las carpetas `certs`, `private` y `newcerts`, crearemos el fichero `index.txt` y el fichero `serial` con un valor de 1000. Para ello introduciremos los siguientes comandos en la terminal:

```
echo 1000 > serial
```

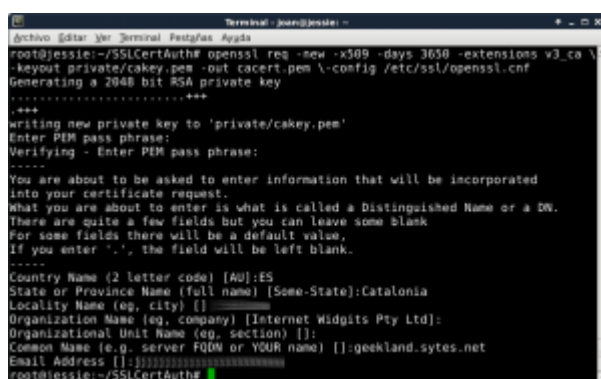
```
touch index.txt
```

Es necesario crear una autoridad de certificación para generar un certificado. Para crear nuestra propia autoridad de certificación tenemos que teclear el siguiente comando en la terminal:

```
openssl req -new -x509 -days 3650 -extensions v3_ca \-keyout private/cakey.pem -
out cacert.pem \-config /etc/ssl/openssl.cnf
```

Al ejecutar el comando lo primero que se nos pregunta es una clave. Esta clave tenemos recordarla y se nos preguntará cada vez que firmemos un certificado. Seguidamente nos pedirá que reconfirmemos la clave que acabamos de elegir.

Seguidamente, tal y como se puede ver en la captura de pantalla, nos irá haciendo preguntas. Las vamos respondiendo una tras otra teniendo especial cuidado en el campo common name que tendrá que ser el fqdn (fully qualified name).



```
Terminal: root@geekland: ~
root@geekland:~/SSLCertAuth# openssl req -new -x509 -days 3650 -extensions v3_ca \-
-keyout private/cakey.pem -out cacert.pem \-config /etc/ssl/openssl.cnf
Generating a 2048 bit RSA private key
.....++++
++++
writing new private key to 'private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Catalonia
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:geekland.sytes.net
Email Address []:
root@geekland:~/SSLCertAuth#
```

Una vez creada la autoridad de certificación, dentro de la ubicación /root/SSLCertAuth podrán encontrar los archivos cakey.pem y cacert.pem. Estos archivos hay conservarlos ya que serán necesarios para crear nuestros certificados.

Ahora creamos una clave privada y la solicitud de firma del certificado (CSR). Para ello introduciremos el siguiente comando en la terminal:

```
openssl req -new -nodes \-out apache-req.pem \-keyout private/apache-key.pem \-
config /etc/ssl/openssl.cnf
```

Después de ejecutar el comando, tal y como se puede ver en la captura de pantalla, se nos irá haciendo preguntas. Las vamos respondiendo una tras otra teniendo especial cuidado en el campo common name que tendrá que ser el fqdn (fully qualified name).

```

Terminal - jessie@jessie: ~
root@jessie:~/SSLCertAuth# openssl req -new -nodes \-out apache-req.pem \-keyout
private/apache-key.pem \-config /etc/ssl/openssl.cnf
Generating a 2048 bit RSA private key
.....+++++
writing new private key to 'private/apache-key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Catalonia
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:geekland.sytes.net
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:solsona
An optional company name []:
root@jessie:~/SSLCertAuth#

```

En la parte final de las preguntas se nos pregunta un password. Este password será para proteger la clave privada que estamos creando. Esta clave privada será usada más adelante en el proceso de autenticación SSL.

Una vez finalizado el proceso en la ubicación /root/SSLCertAuth dispondremos de los siguientes archivos:

*apache-req.pem*: Es el archivo que contiene la solicitud de firma del certificado.

*apache-key.pem*: Es la clave privada que hemos generado. Está clave privada se usa para el proceso de autenticación SSL de los usuarios que se conectan al servidor de owncloud.

Finalmente tan solo hay que generar nuestro propio certificado autofirmado. Para ello tecleamos el siguiente comando en la terminal:

```
openssl ca \-config /etc/ssl/openssl.cnf \-out apache-cert.pem \-infile apache-req.pem
```

Al ejecutar el comando se nos preguntará la clave que fijamos en el apartado de crear una autoridad de certificación. La introducimos y presionamos Enter. Seguidamente se nos preguntará si queremos firmar el certificado e introducirlo en la base de datos. En ambos casos respondemos que Sí. Después de realizar estos pasos, tal y como se puede ver en la captura de pantalla, el proceso ha terminado:



`<Directory /var/www/html/owncloud>` (usar ruta de instalación de owncloud)

`AllowOverride All`

`</Directory>`

`<Directory /home/datosowncloud>` (usar ruta de almacenamiento de datos de Owncloud)

`AllowOverride All`

`</Directory>`

`SSLEngine on`

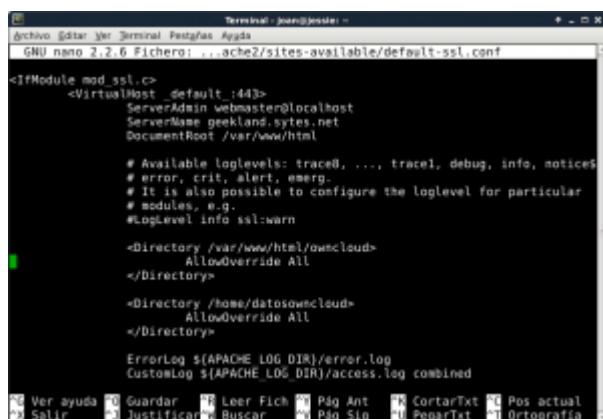
`SSLCertificateFile /etc/ssl/crt/apache-cert.pem` (usar la ruta de certificado)

`SSLCertificateKeyFile /etc/ssl/key/apache-key.pem` (usar la ruta de vuestra clave privada)

`ErrorLog ${APACHE_LOG_DIR}/error_log` (usar la ruta que queráis para almacenar los logs)

`CustomLog ${APACHE_LOG_DIR}/access.log combined` (usar la ruta que queráis para almacenar los logs)

Una vez aplicados los cambios el archivo default-ssl.conf deberá tener un aspecto parecido al siguiente:



```

Terminal: zoran@oside: ~
gchivo [editar] [ver] [terminal] [ pestañas ] [ ayuda ]
GNU nano 2.2.6 Fichero: ...ache2/sites-available/default-ssl.conf
<!--IfModule mod_ssl.c>
<VirtualHost default :443>
ServerAdmin webmaster@localhost
ServerName geekland.bytes.net
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notices
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

<Directory /var/www/html/owncloud>
    AllowOverride All
</Directory>

<Directory /home/datosowncloud>
    AllowOverride All
</Directory>

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
  
```

Si el contenido de la captura de la pantalla es similar tan solo tenemos que guardar los cambios y cerrar el fichero.

Seguidamente crearemos un enlace simbólico del archivo `/etc/apache2/sites-available/default-ssl.conf` en la ubicación `/etc/apache2/sites-enabled/`. Para ello ejecutamos el siguiente comando en la terminal:

`ln -s /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-enabled/default-ssl.conf`

Una vez creado el enlace simbólico, el servicio https está activado. Finalmente tan solo tenemos que reiniciar el servidor apache ejecutando el siguiente comando en la terminal:

```
service apache2 restart
```

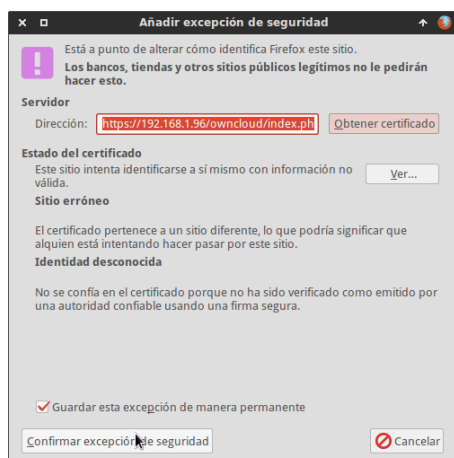
En estos momentos ya podemos intentar acceder a owncloud mediante https. Para ello en un equipo perteneciente a nuestra red local, abrimos el navegador y tecleamos la siguiente url:

```
https://192.168.10.1/owncloud
```

Después de acceder la URL indicada nos aparecerá la advertencia que la conexión no está verificada:

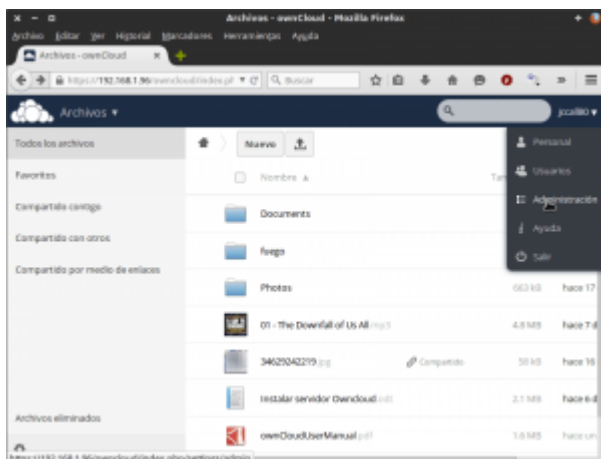


Esta advertencia es completamente normal. El motivo de la advertencia es porque hemos sido nosotros mismos quien ha creado y firmado el certificado ssl, y obviamente nosotros no somos una autoridad certificadora verificada/reconocida. Lo único que tenemos hacer para solucionar este problema es clicar en el apartado Entiendo los riesgos, y seguidamente clicar en el apartado Añadir excepción. Después de clicar en Añadir excepción aparecerá la siguiente ventana:

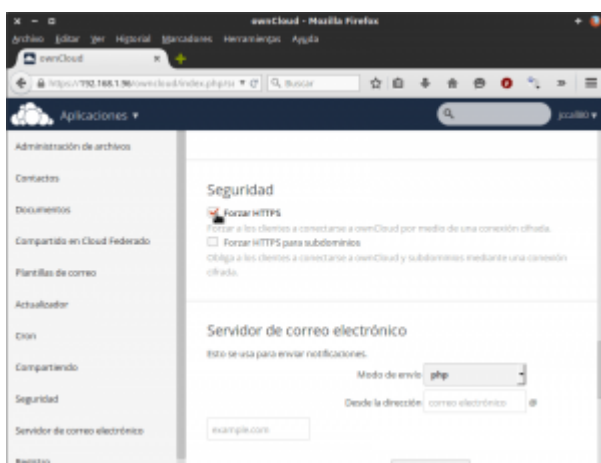




Presionamos encima del botón Confirmar excepción de seguridad y ya podremos acceder a nuestro servidor Owncloud. Una vez dentro de nuestro servidor Owncloud, tal y como se puede ver en la captura de pantalla, accedemos al menú de Administración:



Una vez dentro del menú de administración, tal y como se puede ver en la captura de pantalla, dentro del apartado de seguridad tenemos que marcar la opción Forzar HTTPS.



Después de marcar esta opción, únicamente se podrá acceder al servidor owncloud mediante https.

## PRUEBAS FINALES

```

pi@rasp-lic-02 ~ $ scp -r python_test 192.168.10.102:/home/pi
The authenticity of host '192.168.10.102 (192.168.10.102)' can't be established.
ECDSA key fingerprint is 8d:85:84:17:0e:02:d5:98:8d:48:9c:09:02:e4:0c:90.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.10.102' (ECDSA) to the list of known hosts.

.md5_attack.py          100% 171      0.2KB/s   00:00
md5_attack.py          100% 479      0.5KB/s   00:00
dict.txt               100% 81KB    81.1KB/s  00:00

```

## Ataque de cifrado md5

```

pi@rasp-lic-01 ~ $ ab -n100 -c100 http://localhost/
This is ApacheBench, Version 2.3 <$Revision: 1604373 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done

Server Software:      lighttpd/1.4.35
Server Hostname:      localhost
Server Port:          80

Document Path:        /
Document Length:      11104 bytes

Concurrency Level:    100
Time taken for tests:  0.331 seconds
Complete requests:    100
Failed requests:       0
Total transferred:    1136300 bytes
HTML transferred:     1110400 bytes
Requests per second:  302.10 [#/sec] (mean)
Time per request:     331.018 [ms] (mean)
Time per request:     3.310 [ms] (mean, across all concurrent requests)
Transfer rate:        3352.29 [Kbytes/sec] received

Connection Times (ms)
          min  mean[+/-sd] median   max
Connect:    55   86  19.1    86   116
Processing:  48   94  35.1    84   160
Waiting:    41   91  37.7    83   160
Total:      159  181  17.1   174   217

Percentage of the requests served within a certain time (ms)
 50%    174
 66%    188
 75%    195
 80%    200
 90%    209
 95%    213
 98%    216
 99%    217
100%    217 (longest request)

```

Ataque de 10000 peticiones para hacer caer el servicio.

```

pi@rasp-lic-01 ~ $ ab -n100 -c101 http://localhost/
ab: Cannot use concurrency level greater than total number of requests
Usage: ab [options] [http[s]://]hostname[:port]/path
Options are:
  -n requests      Number of requests to perform
  -c concurrency   Number of multiple requests to make at a time
  -t timelimit     Seconds to max. to spend on benchmarking
                  This implies -n 50000
  -s timeout       Seconds to max. wait for each response
                  Default is 30 seconds
  -b windowsize    Size of TCP send/receive buffer, in bytes
  -B address       Address to bind to when making outgoing connections
  -p postfile      File containing data to POST. Remember also to set -T
  -u putfile       File containing data to PUT. Remember also to set -T
  -T content-type  Content-type header to use for POST/PUT data, eg.
                  'application/x-www-form-urlencoded'
                  Default is 'text/plain'
  -v verbosity    How much troubleshooting info to print
  -w              Print out results in HTML tables
  -i              Use HEAD instead of GET
  -x attributes    String to insert as table attributes
  -y attributes    String to insert as tr attributes
  -z attributes    String to insert as td or th attributes
  -C attribute     Add cookie, eg. 'Apache=1234'. (repeatable)
  -H attribute     Add Arbitrary header line, eg. 'Accept-Encoding: gzip'
                  Inserted after all normal header lines. (repeatable)
  -A attribute     Add Basic WWW Authentication, the attributes
                  are a colon separated username and password.

```

Servicio caído con 10001 peticiones

```
pi@rasp-lic-01 ~ $ sysbench --test=fileio --file-total-size=5G --file-test-mode=rndrw --init-
rng=on --max-time=300 --max-requests=0 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1
Initializing random number generator from timer.

Extra file open flags: 0
128 files, 40Mb each
5Gb total file size
Block size 16Kb
Number of random requests for random IO: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Threads started!
FATAL: Failed to read file! file: 104 pos: 12304384 errno = 0 ()
Done.

Operations performed: 0 Read, 0 Write, 0 Other = 0 Total
Read 0b Written 0b Total transferred 0b (0b/sec)
0.00 Requests/sec executed

Test execution summary:
total time: 0.0076s
total number of events: 0
total time taken by event execution: 0.0000
per-request statistics:
  min: 18446744073709.55ms
  avg: 0.00ms
  max: 0.00ms

Threads fairness:
events (avg/stddev): 0.0000/0.00
execution time (avg/stddev): 0.0000/0.00
```

Prueba de envío de archivo de 5GB sin problemas