

**UNIVERSIDAD NACIONAL DEL ALTIPLANO**

**FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,  
ELECTRÓNICA Y SISTEMAS**

**ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA**



**“DISEÑO DE UN SISTEMA DE ADQUISICIÓN Y TRANSMISIÓN DE  
SONIDOS RESPIRATORIOS UTILIZANDO TELEFONÍA IP”**

**TESIS**

**PRESENTADO POR:**

**FREDY CHARCA MAMANI**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:**

**INGENIERO ELECTRÓNICO**

**PUNO**

**PERÚ**

**2014**

UNIVERSIDAD NACIONAL DEL ALTIPLANO

FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA, ELECTRÓNICA Y  
SISTEMAS

CARRERA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

“DISEÑO DE UN SISTEMA DE ADQUISICIÓN Y TRANSMISIÓN DE  
SONIDOS RESPIRATORIOS UTILIZANDO TELEFONÍA IP”

TESIS PRESENTADA POR:

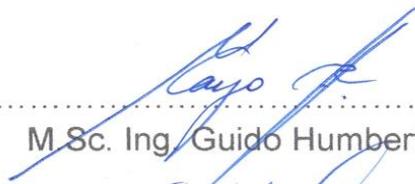
**FREDY CHARCA MAMANI**

PARA OPTAR EL TÍTULO PROFESIONAL DE:

**INGENIERO ELECTRÓNICO**

APROBADA POR EL JURADO REVISOR CONFORMADO POR:

PRESIDENTE:



M.Sc. Ing. Guido Humberto Cayo Cabrera

PRIMER MIEMBRO:



Mg. Ing. Midwar Elías Valencia Vilca

SEGUNDO MIEMBRO:



Ing. Marco Antonio Ramos Gonzáles

DIRECTOR DE TESIS:



M.Sc. Ing. Pedro Bejar Muñoz

ASESOR DE TESIS:



Ing. Ferdinand Edgardo Pineda Ancco

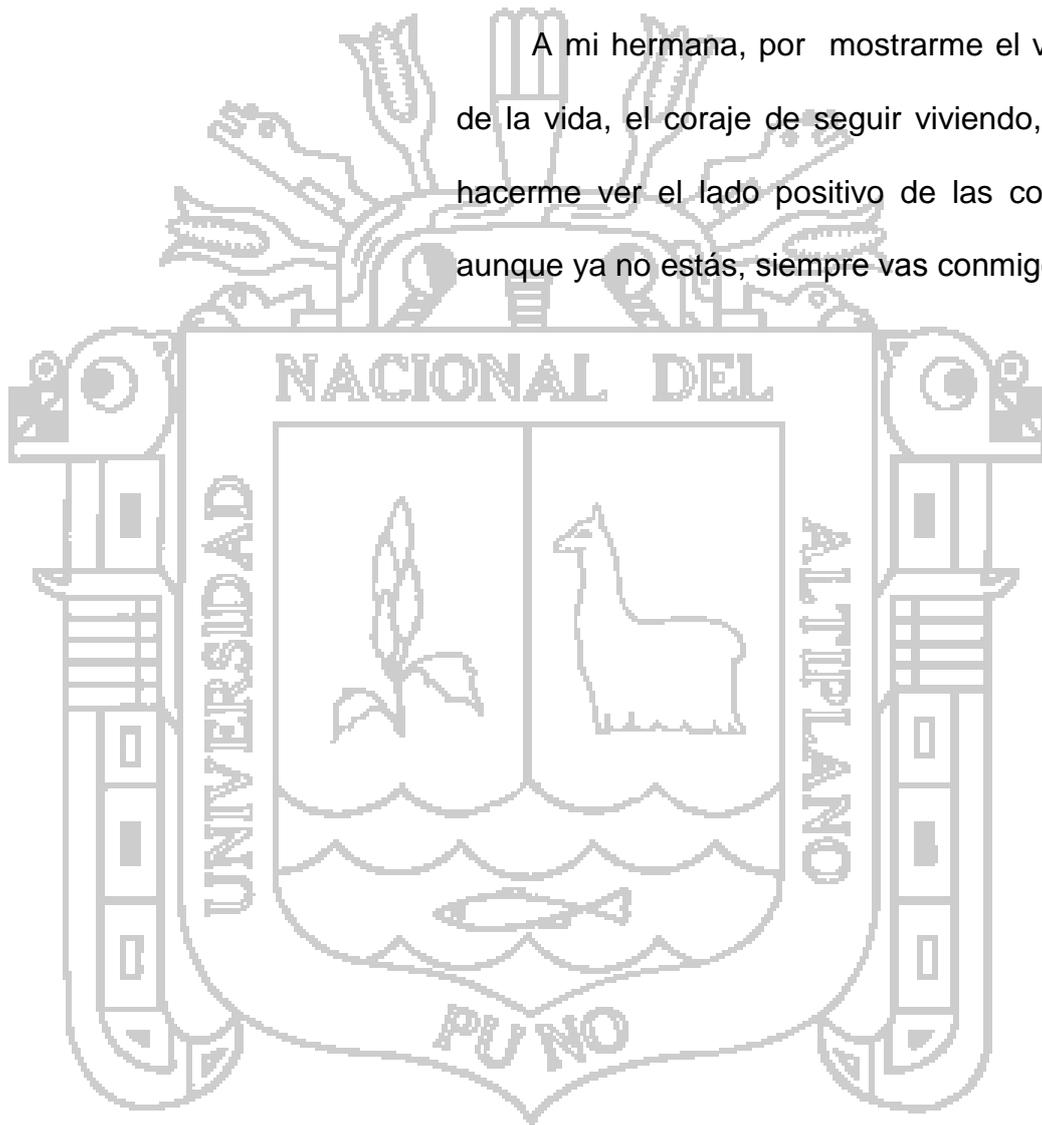
PUNO

PERÚ

2014

## DEDICATORIA

A mi hermana, por mostrarme el valor de la vida, el coraje de seguir viviendo, por hacerme ver el lado positivo de las cosas, aunque ya no estés, siempre vas conmigo...



## AGRADECIMIENTOS

Me gustaría dar las gracias en primer lugar a mis docentes de Ing. Electrónica, por sus esfuerzos y por permitirme divagar y dar riendas sueltas a un inquieto conocimiento y que he intentado plasmar aquí. Al Ing. Ferdinand, por darme un primer salto al mundo del software libre, y siempre haber confiado en mí, Ing. Guido Humberto Cayo Cabrera por estar siempre dispuesto a ayudar y darme esos consejos en los momentos que más lo necesité.

A mis padres por su espera, su silencio, su esfuerzo, sus deseos y aliento de superación. A ellos les debo todo.

A mi hermano, por su esfuerzo, siempre va un paso adelante. A mi hermana, por su positividad y por su paciencia infinita.

A Tania, por su empuje, por creer en mí, por estar siempre dispuesta a ayudar. Sin su tenacidad no habría llegado hasta aquí.

Y por último un sincero agradecimiento a todas las personas que de una u otra forma, brindaron su colaboración para la realización de este trabajo

## ÍNDICE:

DEDICATORIA.....	I
AGRADECIMIENTOS .....	II
ÍNDICE: .....	III
ÍNDICE DE FIGURAS: .....	VI
ÍNDICE DE CUADROS: .....	VIII
RESUMEN.....	1
ABSTRACT .....	2
INTRODUCCIÓN .....	3
<b>CAPITULO I: PLANTEAMIENTO DEL PROBLEMA, ANTECEDENTES Y OBJETIVOS DE LA INVESTIGACIÓN .....</b>	<b>4</b>
<b>1.1. PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>5</b>
<b>1.2. FORMULACIÓN DEL PROBLEMA.....</b>	<b>6</b>
1.2.1. Caracterización del Problema.....	6
1.2.2. Sistematización del Problema .....	6
<b>1.3. ANTECEDENTES.....</b>	<b>7</b>
<b>1.4. JUSTIFICACIÓN .....</b>	<b>8</b>
<b>1.5. OBJETIVOS DE ESTUDIO.....</b>	<b>8</b>
1.5.1. Objetivo General .....	8
1.5.2. Objetivo Especifico.....	8
<b>CAPITULO II. MARCO TEÓRICO, MARCO CONCEPTUAL E HIPOTESIS DE LA INVESTIGACIÓN .....</b>	<b>9</b>
<b>2.1. MARCO TEÓRICO .....</b>	<b>10</b>
2.1.1. Descripción del Sistema Respiratorio Humano.....	10
2.1.1.1. Pulmones.....	11
2.1.1.2. Conductos de las Vías Aéreas .....	12
2.1.1.3. Alvéolos .....	12
2.1.1.4. Circulación Pulmonar.....	13
2.1.1.5. Músculos Respiratorios.....	14

2.1.2. Generación de los Sonidos por Cavidades Resonantes y Física del Sistema Respiratorio.....	17
2.1.2.1. Sonidos Respiratorios Normales.....	20
2.1.2.2. Sonidos Anormales o Adventicios.....	21
2.1.3. Los sonidos respiratorios y su uso como herramienta diagnóstica .....	24
2.1.4. Enfermedades respiratorias comunes .....	26
2.1.5. Filtros .....	28
2.1.5.1. Filtro Pasa Baja.....	29
2.1.5.2. Filtro Pasa Altas.....	30
2.1.5.3. Filtro Pasa Banda .....	30
2.1.5.4. Filtro rechaza Banda.....	31
2.1.5.5. Características de la Respuesta de un Filtro .....	34
2.1.6. Red de Telefonía IP.....	36
2.1.6.1. Introducción .....	36
2.1.6.2. Protocolos Relacionados con VoIP .....	38
2.1.6.3. Protocolos de Control y Señalización de Llamadas.....	42
2.1.6.4. Calidad.....	44
2.1.7. Asterisk .....	48
2.1.7.1. Introducción .....	48
2.1.7.2. Arquitectura de Asterisk .....	49
<b>2.2. MARCO CONCEPTUAL.....</b>	<b>50</b>
2.2.1. Sonidos respiratorios.....	50
2.2.2. Auscultación pulmonar .....	50
2.2.3. Estetoscopio.....	51
2.2.4. Asterisk .....	51
<b>2.3. HIPÓTESIS Y VARIABLES.....</b>	<b>52</b>
2.3.1. Formulación de la hipótesis.....	52
2.3.2. Operacionalización de las variables .....	52
2.3.3. Contraste de Hipótesis .....	52
<b>CAPITULO III. METODOLOGÍA DE INVESTIGACIÓN.....</b>	<b>53</b>
<b>3.1. DESCRIPCIÓN DEL MÉTODO Y DISEÑO.....</b>	<b>54</b>
3.1.1. Tipo de Investigación.....	54
3.1.2. Ámbito de Estudio .....	54
3.1.3. Nivel de Investigación.....	54
3.1.4. Administración del Proyecto .....	54
3.1.4.1. Recursos humanos .....	54
3.1.4.2. Presupuesto.....	55
3.1.4.3. Financiamiento.....	55
<b>CAPITULO IV. DISEÑO, IMPLEMENTACIÓN Y ANÁLISIS DE LA INGENIERÍA DEL PROYECTO .....</b>	<b>56</b>
<b>4.1. CIRCUITO DE ADQUISICIÓN DE SONIDOS RESPIRATORIOS.....</b>	<b>57</b>
4.1.1. Descripción general.....	57

4.1.2. Sensor.....	58
4.1.3. Amplificador de Alta Impedancia .....	59
4.1.4. Filtros Activos.....	61
4.1.4.1. Filtro Paso Alto.....	62
4.1.4.2. Filtro Paso Bajo.....	64
4.1.5. Ajustador de Nivel .....	66
4.1.5.1. Diagrama Interno del INA129.....	67
4.1.6. Diseño Completo.....	70
<b>4.2. Medio de Transmisión .....</b>	<b>72</b>
4.2.1. Instalación de Linux Debian Wheesy.....	72
4.2.2. Instalación de Asterisk.....	83
4.2.3. Configuración de Asterisk.....	91
4.2.3.1. Configurando el protocolo DUNDi.....	92
4.2.3.2. Configurando el protocolo SIP .....	114
<b>CONCLUSIONES.....</b>	<b>127</b>
<b>RECOMENDACIONES.....</b>	<b>128</b>
<b>BIBLIOGRAFÍA Y REFERENCIAS .....</b>	<b>129</b>
<b>GLOSARIO DE TÉRMINOS MÉDICOS.....</b>	<b>130</b>
<b>ANEXOS .....</b>	<b>132</b>

## ÍNDICE DE FIGURAS:

<b>Figura 1.</b> Episodios por neumonías en menores de 5 años Perú .....	5
<b>Figura 2.</b> Esquema de los pulmones humanos.....	12
<b>Figura 3.</b> La estructura de los alvéolos.....	13
<b>Figura 4.</b> Esquema de las etapas de la respiración.....	16
<b>Figura 5.</b> Hallazgos percutórios normales en las distintas áreas topográficas del tórax. .....	19
<b>Figura 6.</b> Ruidos adventicios .....	24
<b>Figura 7.</b> Diagramas de un filtro pasa baja ideal .....	29
<b>Figura 8.</b> Diagrama de un filtro pasa alta ideal .....	30
<b>Figura 9.</b> Diagrama de un filtro pasa banda ideal .....	31
<b>Figura 10.</b> Diagrama de un filtro rechaza banda ideal .....	31
<b>Figura 11.</b> Resultado de un filtro pasa banda.....	32
<b>Figura 12.</b> Diagrama de bloques de filtro pasa banda .....	32
<b>Figura 13.</b> Diagrama de bloques de filtro rechaza banda .....	33
<b>Figura 14.</b> Resultado de un filtro rechaza banda.....	33
<b>Figura 15.</b> Gráfica comparativa de tres tipos de características de respuesta de filtros .....	34
<b>Figura 16.</b> Comparación de la respuesta a la frecuencia de tres filtros pasa altas Butterworth .....	36
<b>Figura 17.</b> Comparación de la respuesta a la frecuencia de tres filtros pasa altas Butterworth .....	36
<b>Figura 18.</b> Modelo de referencia OSI .....	37
<b>Figura 19.</b> Encapsulamiento VoIP .....	37
<b>Figura 20.</b> Cabecera RTP .....	40
<b>Figura 21.</b> Tamaño de cabeceras IP/UDP/RTP.....	41
<b>Figura 22.</b> Variación de retardo.....	46
<b>Figura 23.</b> Arquitectura de Asterisk .....	49
<b>Figura 24.</b> Diagrama a bloques del circuito de adquisición de sonidos respiratorios ..	57
<b>Figura 25.</b> Primera parte del estetoscopio digital.....	58
<b>Figura 26.</b> Esquemático del micrófono .....	59
<b>Figura 27.</b> Montaje amplificador inversor.....	59
<b>Figura 28.</b> Simulación Experimental del Amplificador 10 Hz.....	61
<b>Figura 29.</b> Diseño del Filtro Paso Alto .....	63
<b>Figura 30.</b> Diseño del Filtro Paso Bajo .....	65
<b>Figura 31.</b> Diagrama Interno INA129P .....	68
<b>Figura 32.</b> Tabla de datos del INA129P.....	69
<b>Figura 33.</b> Diseño del Ajustador de Nivel .....	70
<b>Figura 34.</b> Diseño Completo.....	71
<b>Figura 35.</b> Ubicación de los componentes en 3D .....	71
<b>Figura 36.</b> Diseño de una Red WAN .....	72
<b>Figura 37.</b> VirtualBOX .....	73
<b>Figura 38.</b> Ejecución de Debian linux .....	73
<b>Figura 39.</b> Selección del lenguaje .....	74
<b>Figura 40.</b> Selección de ubicación.....	74

<b>Figura 41.</b> Selección de teclado .....	75
<b>Figura 42.</b> Reconocimiento del Hardware .....	75
<b>Figura 43.</b> Nombre de la maquina .....	75
<b>Figura 44.</b> Nombre de dominio .....	76
<b>Figura 45.</b> Clave de usuario .....	76
<b>Figura 46.</b> Nombre de usuario.....	77
<b>Figura 47.</b> Contraseña de usuario.....	77
<b>Figura 48.</b> Particionado de disco guiado .....	77
<b>Figura 49.</b> Particionado de disco.....	78
<b>Figura 50.</b> Particionado de disco guiado recomendado para novatos .....	78
<b>Figura 51.</b> Finalizar particionado de disco guiado .....	78
<b>Figura 52.</b> Confirmar particionado de disco guiado .....	79
<b>Figura 53.</b> Instalando el sistema base.....	79
<b>Figura 54.</b> Configurando gestor de paquetes .....	79
<b>Figura 55.</b> Configurando réplica de red.....	80
<b>Figura 56.</b> Configurando réplica de red por el usuario.....	80
<b>Figura 57.</b> Configuración de popularidad.....	81
<b>Figura 58.</b> Selección de programas a instalar .....	81
<b>Figura 59.</b> Instalando programas del sistema operativo .....	81
<b>Figura 60.</b> Configuración de GRUB.....	82
<b>Figura 61.</b> Finalizando la instalación .....	82
<b>Figura 62.</b> Inicio del sistema operativo instalado .....	82
<b>Figura 63.</b> Ejecutando terminal como superusuario .....	83
<b>Figura 64.</b> Configurando el fichero sources.list.....	85
<b>Figura 65.</b> Configuración de Core Sound Packages.....	89
<b>Figura 66.</b> Esquema de conexión de los servidores DUNDi .....	93
<b>Figura 67.</b> Comprobación del registro DUNDi en el servidor A .....	112
<b>Figura 68.</b> Comprobación del registro DUNDi en el servidor B.....	113
<b>Figura 69.</b> Comprobación del registro DUNDi en el servidor C.....	113
<b>Figura 70.</b> Esquema de conexión de los servidores SIP .....	114
<b>Figura 71.</b> Comprobación del registro SIP en el servidor A.....	123
<b>Figura 72.</b> Comprobación del registro SIP en el servidor B.....	123
<b>Figura 73.</b> Teléfono IP LINKSYS SPA 841 .....	124
<b>Figura 74.</b> Sofphone CounterPatch.....	125
<b>Figura 75.</b> Llamando de servidor A a servidor B.....	125
<b>Figura 76.</b> Aceptando llamada .....	126
<b>Figura 77.</b> Comunicación de terminales servidor A y servidor B.....	126
<b>Figura 78.</b> Prueba del circuito en protoboard .....	141
<b>Figura 79.</b> Ensamble del circuito .....	141
<b>Figura 80.</b> Compilando Asterisk .....	142
<b>Figura 81.</b> Prueba de servidores Asterisk.....	142
<b>Figura 82.</b> Funcionamiento del proyecto .....	143

## ÍNDICE DE CUADROS:

<b>Tabla I:</b> Tipos de sonidos respiratorios y sus características acústicas principales ....	23
<b>Tabla II:</b> Patologías más frecuentes .....	28
<b>Tabla III:</b> Operacionalización de las variables.....	52
<b>Tabla IV:</b> Costo de herramientas y materiales para la propuesta .....	55



## RESUMEN

En este trabajo se describe el desarrollo de un sistema de adquisición de sonidos respiratorios mediante un estetoscopio de bajo coste y un sistema de transmisión extremo-extremo del audio adquirido. El sistema se constituye de circuitería y programación. La circuitería está integrada por un micrófono con campana (acoplado por aire), un amplificador inversor, una etapa de filtraje analógico y otra etapa final para ajustar el nivel del audio deseado a la salida del estetoscopio un amplificador de instrumentación. El circuito de conformación de señal está entre las frecuencias de 60Hz y de 4KHz, lo cual asegura la adecuada adquisición de los sonidos respiratorios. Este dispositivo es conectado a un ordenador, para la adquisición y el procesamiento de los datos. La programación para la transmisión del sonido respiratorio a través de la red de internet está desarrollada en Asterisk. Asterisk es una central de telefonía IP "open source" que cumple con todas las funcionalidades de una Central telefónica PBX, para su funcionamiento del prototipo, previamente se instaló el software libre Debian en tres ordenadores y utilizando sus Protocolos de control y señalización mediante SIP y DUNDi con sus respectivos códec se logró la comunicación y transmisión de los sonidos respiratorios para fines de diagnóstico del paciente, remotamente.

**Palabras claves:** Sonido respiratorio, Asterisk, Licencia GPL, Protocolo SIP y DUNDi.

## ABSTRACT

In this paper the development of an acquisition system described breath sounds with a stethoscope low cost and a transmission system acquired end - end audio. The system is structured as circuitry and programming. The circuitry consists of a microphone with hood (air coupled), an inverting amplifier, analog filtering stage and final stage to adjust the desired level at the output of an instrumentation amplifier stethoscope audio. The signal shaping circuit is between the frequencies of 60Hz and 4kHz, which ensures the proper acquisition of respiratory sounds. This device is connected to a computer, for the acquisition and processing of data. Programming for respiratory sound transmission through the internet network is developed in Asterisk. Asterisk in a central IP telephony "open source" to comply with all the functionality of a telephone PBX Central, for operation of the prototype, pre Debian free software was installed on three computers and using their control and signaling protocols by SIP and DUNDi codec with their communication and transmission of breath sounds for the purpose of patient diagnosis, remotely managed.

**Keywords:** Sound respiratory, Asterisk, GPL license, SIP protocol and DUNDi.

## INTRODUCCIÓN

La presente tesis titulada “Diseño de un sistema de adquisición y transmisión de sonidos respiratorios utilizando telefonía IP”, se realizó teniendo en cuenta que la auscultación de los sonidos respiratorios presenta grandes ventajas como procedimiento de diagnóstico clínico, ya que por su facilidad de realización y ausencia de riesgos para la salud de las personas es factible de practicar en cualquier paciente, independientemente de su edad y estado de salud. El análisis está realizado con frecuencia con un pico máximo por debajo de los 4KHz, en donde sus características se mezclan con las características en frecuencia de sonidos musculares y cardiovasculares. Por lo tanto, los sonidos respiratorios no son fácilmente distinguibles de estos sonidos de interferencia.

La presente tesis está organizada en cuatro capítulos:

El capítulo I referido al problema en que se aborda la descripción y las causas del problema bajo; el enfoque de la metodología del estructural lógico, así mismo, trata los antecedentes, la justificación, y los objetivos de la delimitación del problema.

El capítulo II referido a los fundamentos técnicos de la investigación, en que se desarrolla el marco teórico, abordándose aspectos relacionados con el estudio de los sonidos respiratorios y la plataforma o medio de transmisión desarrollado. Por otro lado se describe el marco conceptual, la finalidad, los objetivos, la hipótesis y las variables de la investigación.

El capítulo III referido la descripción general del proyecto; tipo de investigación, ámbito de estudio y la administración del proyecto.

Finalmente el capítulo IV aborda la ingeniería del proyecto; el diseño, análisis y la implementación de la circuitería de adquisición de sonidos respiratorios, el desarrollo del medio de transmisión mediante Linux y la programación del software Asterisk de forma detallada, y por último se hacen las pruebas correspondientes (trunk), así mismo desarrolla las conclusiones, recomendaciones, y bibliografía.



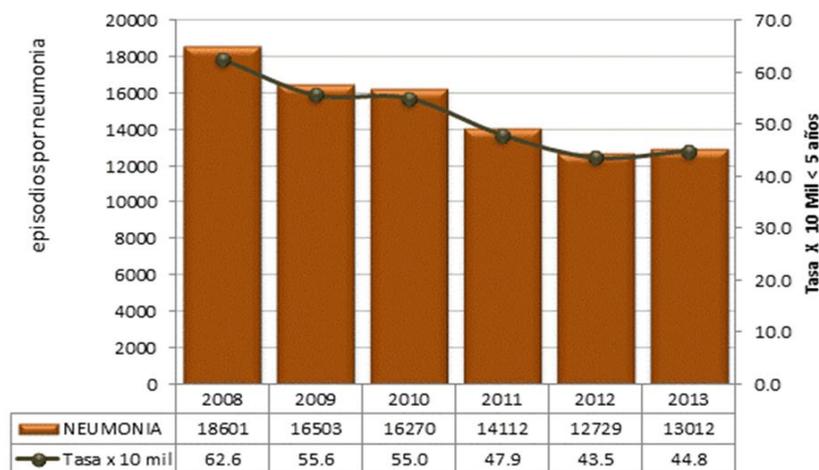
**CAPITULO I: PLANTEAMIENTO DEL PROBLEMA,  
ANTECEDENTES Y OBJETIVOS DE LA INVESTIGACIÓN**

### 1.1. PLANTEAMIENTO DEL PROBLEMA

Las infecciones respiratorias agudas son responsables de la muerte de cerca de 4 millones de niños por año, principalmente en países en vías de desarrollo como son Asia, África y América Latina. La neumonía es responsable de aproximadamente el 85% de todas las muertes por IRA, pero también contribuyen a esta mortalidad la bronquiolitis, crup y complicaciones de las vías respiratorias superiores.

En el Perú, Las infecciones respiratorias agudas (IRA) y su complicación más grave, la neumonía, son consideradas como las principales causas de mortalidad en los niños menores de 5 años, con un 20.84% de mil nacidos en el 2010 aunque la cifra bajó con respecto al 2006 a un 57% de mil nacidos, En el 2012, se registraron 2'794,552 casos de infecciones respiratorias agudas y 367 defunciones en menores de 5 años, siendo las regiones de Loreto, Puno, Junín y Cusco las más afectadas. Le siguió Lima, Huánuco, Ucayali, Cajamarca, Arequipa, Pasco, entre otros.

**Episodios por neumonías en menores de 5 años Perú 2008\* – 2013\***



\* Hasta la SE 20  
Fuente: Dirección General de Epidemiología

**Figura 1. Episodios por neumonías en menores de 5 años Perú**

Durante su hospitalización, existe una necesidad de supervisar constantemente la evolución del estado de salud; lamentablemente, en la mayoría de casos ocurridos, los centros de salud de la región presentan el inconveniente de falta de médicos generales y especialistas, escasez de recursos médicos y paramédicos, incremento de pacientes con enfermedades pulmonares en épocas de friaje, centralización de personal capacitado y equipos médicos de última generación tecnológica emergente en los principales nosocomios de la capital.

Usualmente personal paramédico y personal afín (como enfermeras) no están capacitados para realizar el diagnóstico de evolución de la enfermedad respiratoria, que masivamente atienden en zonas rurales y/o marginales distantes de la región; de este modo tal ausentismo de personal médico especialista constituye un factor crítico para la atención de los pacientes.

## **1.2. FORMULACIÓN DEL PROBLEMA**

### **1.2.1. Caracterización del Problema**

¿Será posible diseñar y desarrollar un sistema de adquisición de sonidos respiratorios utilizando telefonía IP para el diagnóstico de patología respiratoria?

### **1.2.2. Sistematización del Problema**

- 1) ¿Qué tecnología se adecuará al diagnóstico de pacientes con patología respiratoria?
- 2) ¿Qué entorno y/o plataforma se adecuará al diseño del sistema con bajo coste y alta performance y eficiencia?
- 3) ¿Cómo se interpretará los resultados para la determinación del diagnóstico del paciente?
- 4) ¿Qué herramientas permitirán el diagnóstico óptimo del paciente?

### 1.3. ANTECEDENTES

En la serie de revistas tecnologías en salud de la SUBSECRETARÍA DE INNOVACIÓN Y CALIDAD CENTRO NACIONAL DE EXCELENCIA TECNOLÓGICA EN SALUD, en el volumen de telemedicina, pagina 13, menciona una de las ventajas más importantes que es la reducción de las desigualdades en la población para tener acceso a los servicios de salud, independientemente de la localización geográfica. En este sentido existen grupos de beneficiados.

En el texto del Dr. WASHIGTON ZARATE ROSPICLIOSI de título “FONOCARDIOGRAFÍA ESTADO DEL ARTE” menciona a uno del los instrumentos utilizados para auscultación en la actualidad que son el resultado de la evolución de este instrumento desde su invención por LAENNEC en 1816 y la auscultación asistida por ordenador mejorara espectacularmente la habilidad del médico para detectar anomalías. Proporcionando acceso inalámbrico a archivos de señales basados en Web, los médicos podrán realizar comparaciones seriadas y remitir de forma ON-LINE del paciente al especialista.

En el texto de Ing. Fabián Selles Rosa de título “INTRODUCCIÓN A LA TELEFONÍA IP UTILIZANDO ESTÁNDARES”, en la página 14 menciona que para poder enviar la voz mediante medios digitales, debemos discretizarla. Para este proceso de discretización o cuantificación se utiliza un códec. Un códec es una caja negra física o lógica, que en la entrada recibe una señal de audio y a la salida devuelve una cadena de bits, o viceversa. Y páginas más abajo menciona los protocolos abiertos de VoIP, que podrían ser utilizados para implementar este proyecto.

En el paper de título “DESARROLLO DE UN SISTEMA DE TELEFONÍA IP DISTRIBUIDO MEDIANTE LA IMPLEMENTACIÓN DE UN MECANISMO DE DESCUBRIMIENTO DE RUTAS DE LLAMADAS, EN BASE AL SISTEMA OPERATIVO LINUX” del Ing. Sánchez Almeida Tarquino, en la página 02 dice que DUNDI es un draft de Internet creado por Mark Spencer y tiene como función el descubrimiento de un número entre diferentes PBXs en una red

punto a punto. Para ello busca y comparte los planes de marcación (dialplan) entre servidores IP-PBX. DUNDI, SIP por consiguiente son protocolos de enrutamiento de VoIP que brinda servicios de directorio similar a los proporcionados por ENUM pero sin la restricción de ser un sistema centralizado.

#### **1.4. JUSTIFICACIÓN**

Las estrategias adecuadas e inmediatas que se puedan implantar en el sistema de atención de pacientes con patología respiratoria a mediano plazo sería vía acceso telemático: interoperabilidad informática y telecomunicaciones en el campo sanitario, juntamente para lograr el objetivo global se complementarían con la adecuada supervisión y/o diagnóstico de los precitados pacientes utilizando la tecnología de telefonía IP y de las redes WAN vía internet.

El diseño basado en el sistema operativo LINUX aplicando Asterisk, vía redes de telefonía IP soportan múltiples protocolos de VoIP, entre ellos DUNDI y SIP que permite conectar varias PBXs adyacentes o remotas a través de una red punto a punto, mediante el enlace de servidores dedicados, logrando la comunicación eficiente y fiable para el diagnóstico remoto del paciente a bajo coste.

#### **1.5. OBJETIVOS DE ESTUDIO**

##### **1.5.1. Objetivo General**

Diseñar y desarrollar un sistema de transmisión remota del sonido respiratorio utilizando telefonía IP para el diagnóstico de patologías respiratorias.

##### **1.5.2. Objetivo Especifico**

- Desarrollar un sistema de transmisión remota para el sonido respiratorio.
- Implementar una red de telefonía IP.
- Desarrollar un instrumento para lograr la adquisición de la señal acústica respiratoria.
- Desarrollar un programa de visualización de las señales acústicas.



## 2.1. MARCO TEÓRICO

### 2.1.1. Descripción del Sistema Respiratorio Humano

La respiración es un proceso fundamental en el ser humano, su función es proporcionar el oxígeno que necesitan las células. Estas consiguen la energía metabolizando glucosa con oxígeno, por lo que requieren un aporte constante de él. El proceso de ventilación consiste en la entrada y salida de aire de los pulmones, la cual permite la obtención de oxígeno y la eliminación del dióxido de carbono de la sangre. De allí el oxígeno se transfiere a la sangre; que lo conduce hasta las células, donde se realiza el metabolismo con la glucosa; producto de estas reacciones se genera el dióxido de carbono que a su vez es transportado en la sangre venosa hasta los pulmones para ser expulsado. Al realizar ejercicio físico se origina un aumento en la ventilación para compensar las mayores demandas de energía de las células musculares.

El proceso por el cual entra el aire a los pulmones se denomina inspiración (inhalación) y el de expulsión del aire se llama espiración (exhalación). La respiración está controlada por el tronco encefálico; no se hace necesario realizar ningún esfuerzo consciente para la inhalación y exhalación del aire, despiertos o dormidos, no se tiene que pensar en respirar, es tan necesario para vivir que se hace automáticamente. Cada día, una persona respira aproximadamente 20.000 veces, y para cuando tenga 70 años habrá ventilado al menos 600 millones de veces, aunque la profundidad y velocidad de la ventilación pueden alterarse voluntariamente.

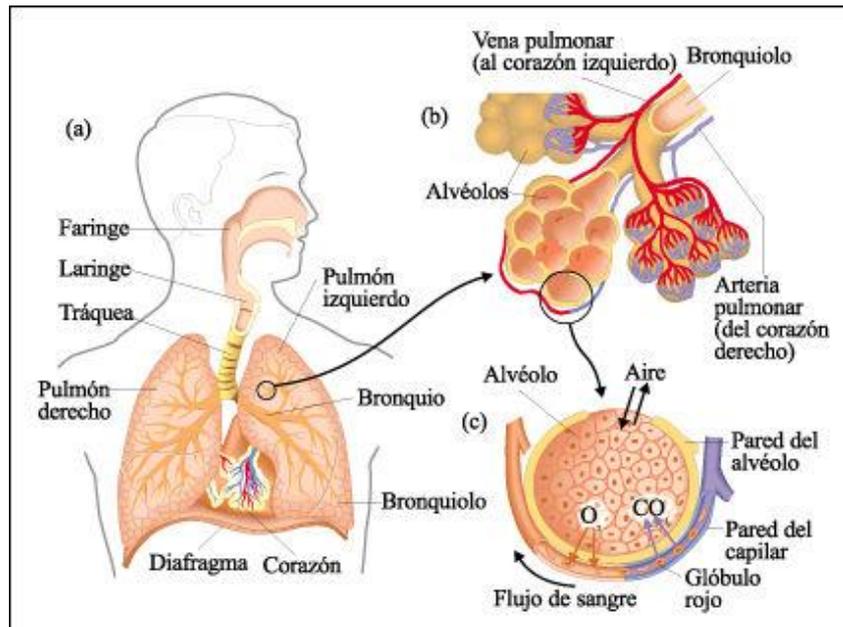
En el proceso de la respiración interviene el sistema respiratorio, el cual incluye pulmones, conductos de las vías aéreas, músculos respiratorios, vasos pulmonares, tejidos y estructuras circundantes; cada uno de ellos juega un papel importante; cuando respiramos, estamos tomando aire rico en oxígeno por la nariz y la boca, los pulmones se llenan (inspiración) y se vacían (espiración), si el aire que respiramos está sucio o contaminado, nuestro sistema inmunológico nos defiende de las sustancias u organismos que entran por la nariz y la boca.

### 2.1.1.1. Pulmones

Como unidades funcionales, el corazón y el pulmón son usualmente considerados como un órgano único complejo, pero un órgano es dedicado a la sangre y el otro para el aire; ellos usualmente se separan en términos de pruebas que conducen a evaluar la función pulmonar o del corazón. El ser humano posee dos pulmones, estos son estructuras elásticas que pueden expandirse o contraerse, mediante el proceso de inspiración y espiración respectivamente, están ubicados en el pecho dentro de la caja torácica, protegidos por las costillas y a ambos lados del corazón; la principal función de los pulmones es la distribución de oxígeno e intercambio de gases. En los pulmones el oxígeno pasa de los alvéolos a los capilares pulmonares y el dióxido de carbono se traslada en sentido opuesto, de los capilares pulmonares al interior de los alvéolos. Esto ocurre simplemente por el fenómeno físico de la difusión (cada gas va de una región donde está más concentrado a otra de menor concentración).

El pulmón derecho consta de tres divisiones llamadas lóbulos y el pulmón izquierdo tiene solo dos, esto se debe a que el corazón ocupa una parte del pecho en la región izquierda lo que disminuye espacio para el pulmón. Son órganos livianos y similares a una esponja debido a que su estructura está compuesta de tubos llenos de aire. En su interior, el pulmón está conformado por las múltiples divisiones que hacen los bronquios hasta la conformación de un lobulillo pulmonar, allí se produce la hematosis, que es el proceso durante el cual los glóbulos rojos absorben oxígeno del aire y liberan el dióxido de carbono hacia el exterior. También están los grandes vasos pulmonares (arterias y venas) que son los encargados de llevar hasta el alvéolo el gas carbónico (arterias pulmonares) y de llevar desde el alvéolo hasta las células el oxígeno necesario para la obtención de energía (venas pulmonares).

En todo el sistema el oxígeno pasa de los pulmones a la sangre y de ésta a los tejidos, de puntos de mayor a menor concentración, hasta ser finalmente utilizado por las células; el dióxido de carbono pasa de las células, donde se produce, a la sangre, a los pulmones y al exterior, siempre hacia zonas de menor concentración (diferencia de presiones).



**Figura 2. Esquema de los pulmones humanos**

**Fuente: Shier -Butler – Lewis**

#### 2.1.1.2. Conductos de las Vías Aéreas

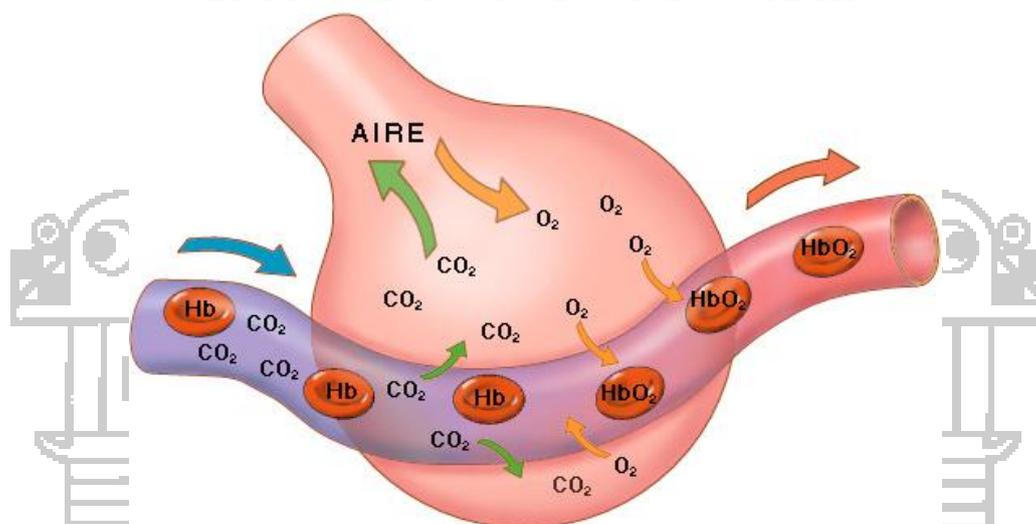
El aire es transportado de la atmósfera a los alvéolos comenzando con las cavidades oral y nasal, dentro de las funciones de las vías aéreas está el calentar del aire inhalado, humidificar el aire completamente a pesar del contacto con la secreción de la mucosa respiratoria y de filtrar el aire que llega a los pulmones, en pocas palabras las vías aéreas son las encargadas de acondicionar el aire.

#### 2.1.1.3. Alvéolos

Los alvéolos son sacos muy pequeños y elásticos, están tan unidos que no tienen una pared independiente en lugar de ello se conforma un tabique interalveolar. Están conformados por dos o tres células epiteliales delgadas, se encuentran en la parte terminal de cada bronquiolo. Su principal función es producir el intercambio gaseoso, es decir, que es aquí donde la sangre va a descargar el gas carbónico expulsado por las células y a recibir el oxígeno que van a necesitar para cumplir con su trabajo.

En la parte interna de cada alvéolo hay células de la sangre, las cuales ingieren y destruyen sustancias irritantes que están en el aire inspirado, de forma tal que cualquier trastorno pulmonar que afecte los alvéolos, provocará una dificultad al respirar.

Una característica funcional determinante de los alvéolos es la permeabilidad que tiene para el paso de los gases arteriales: el oxígeno y el gas carbónico. Esta es la razón por la cual, a esta parte del sistema respiratorio, se le conoce como sistema de intercambio.



**Figura 3. La estructura de los alvéolos**

**Fuente: Shier -Butler – Lewis**

#### **2.1.1.4. Circulación Pulmonar**

La circulación pulmonar es la porción del sistema circulatorio que lleva sangre desoxigenada desde el corazón hasta los pulmones, para luego regresarla oxigenada de vuelta al corazón. Adicionalmente, la circulación pulmonar tiene otras funciones secundarias tales como servir de filtro para la sangre venosa, reteniendo mecánicamente o por adherencia específica, células sanguíneas envejecidas, micro coágulos, células adiposas, células placentarias, etc., también se comporta como un órgano metabólico ya que cumple con la función de nutrir a los tejidos pulmonares, proporcionando los substratos necesarios para sus requerimientos metabólicos y además la circulación pulmonar sirve

como depósito de sangre. Su estructura es adecuada para esta función, ya que su amplia red capilar cubre el 85% de la superficie alveolar, de tal forma que expone la sangre al aire alveolar en una fina capa.

El circuito pulmonar empieza en la aurícula derecha, donde llega prácticamente toda la sangre venosa del organismo, pasa al ventrículo derecho y desde allí es impulsada al territorio alveolar a través de la arteria pulmonar. Una vez arterializada, la sangre es llevada por las venas pulmonares a la aurícula izquierda, donde se incorpora al circuito mayor.

A diferencia de la circulación sistémica (envía la sangre desde el corazón a todas las partes de nuestro cuerpo y después traerla al corazón desoxigenada) la circulación pulmonar tiene un alto flujo, un régimen de presiones menores, debido a su baja resistencia.

#### **2.1.1.5. Músculos Respiratorios**

Las actividades físicas requieren energía, la mayor parte esta energía que se consume durante la actividad física es generada por la acción de los músculos respiratorios, cuya función principal es la de movilizar un volumen de aire adecuado para realizar un intercambio gaseoso apropiado, aportando oxígeno a los diferentes tejidos. Dentro de los músculos respiratorios se tienen:

##### **2.1.1.5.1. Diafragma**

Está ubicado justo en la parte inferior de los pulmones; cuando el diafragma se contrae, la cavidad torácica se ensancha reduciendo la presión interior. Para igualar la presión, el aire entra en los pulmones. En la etapa de inspiración, el diafragma, que tiene forma de cúpula mientras esta en reposo, se tensa y se aplana. Los músculos intercostales se contraen y desplazan la caja torácica hacia arriba y hacia fuera. Estos dos movimientos aumentan el volumen del tórax y provocan la expansión de los pulmones y la entrada de aire en ellos. En la espiración, el diafragma y los músculos inspiratorios del tórax se relajan, mientras los músculos espiratorios se contraen que provocan el descenso de la caja torácica y la compresión de los pulmones, con la consiguiente salida de aire. Los pulmones no se llenan completamente durante la inspiración, ni se

vacían del todo durante la espiración. Con una espiración tranquila y normal solo se exhala una décima parte del aire de los pulmones, aire que es reemplazado por una cantidad equivalente de aire fresco. Este aire nuevo (aire respiratorio) se mezcla con el viejo (volumen residual) que hay en los pulmones.

#### **2.1.1.5.2. Músculos de la caja torácica**

Los músculos más importantes que actúan sobre la caja torácica son los intercostales. Se trata de músculos que se posicionan en dos finos planos de fibras musculares ocupando los espacios intercostales, y denominándose externos o internos en función de su relación con la superficie torácica.

##### **2.1.1.5.2.1. Los Músculos Intercostales Externos**

Son los que levantan las costillas para aumentar el volumen de la cavidad torácica y así producir la inspiración. Están posicionados entre las costillas y ayuda a la inspiración por el movimiento de las costillas arriba y hacia delante. Esto, entonces, incrementa el volumen del tórax. Los intercostales internos tienen unas fibras que van oblicuamente en dirección caudal y, posteriormente, desde la costilla superior a la inferior.

La función de la respiración se puede describir en seis etapas (ver figura 4) como se enuncia a continuación:

**Etapas 1.** Aquí el aire conteniendo oxígeno es inspirado al interior de los pulmones y penetra en los alvéolos. El oxígeno se pasa desde el aire al interior de los vasos sanguíneos que rodean los alvéolos.

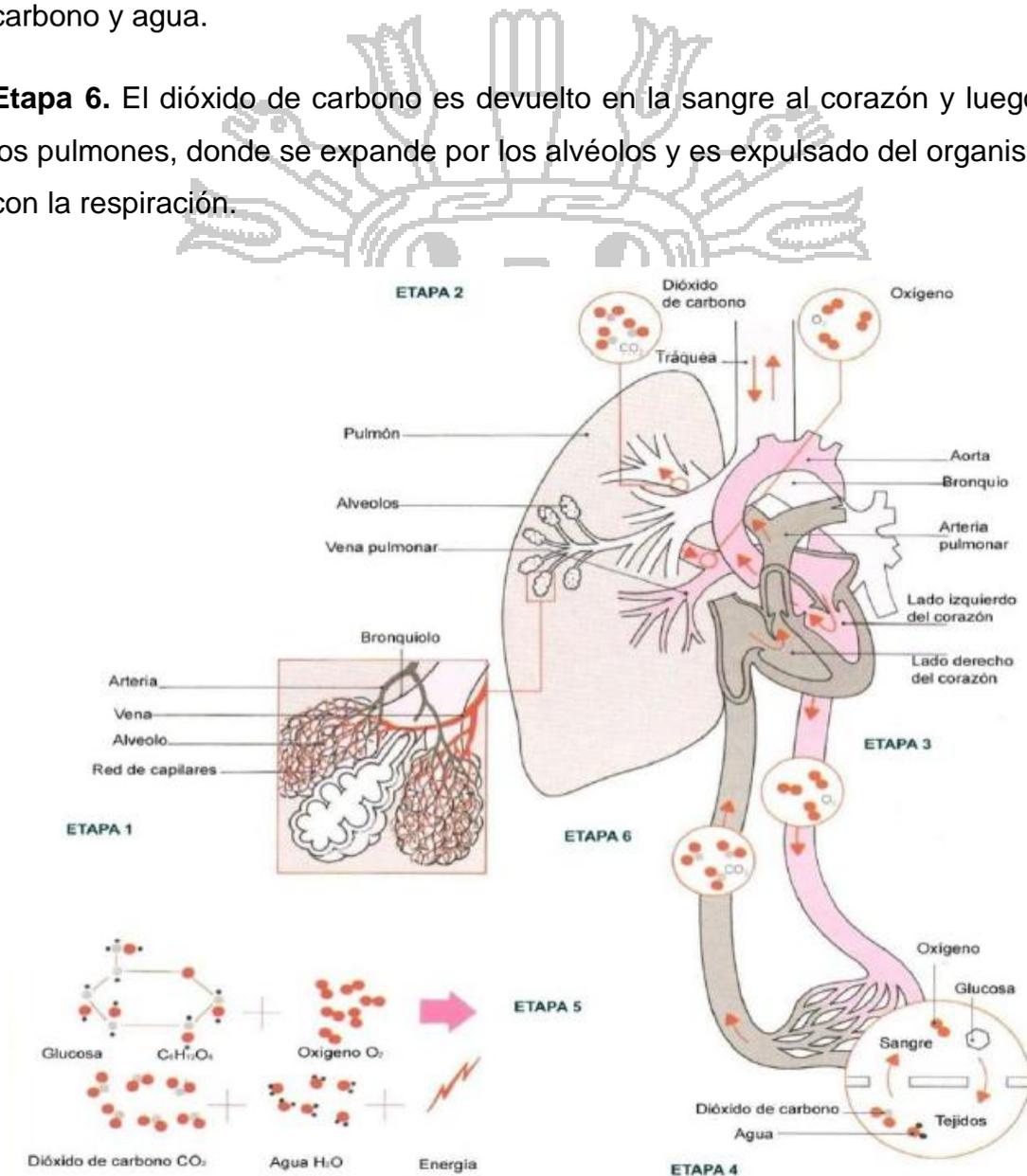
**Etapas 2.** La sangre saturada de oxígeno pasa desde los pulmones, por las venas pulmonares, hasta el lado izquierdo del corazón.

**Etapas 3.** Desde el lado izquierdo del corazón, la sangre oxigenada es bombeada a través de la aorta hasta los tejidos corporales. El oxígeno es transportado en la sangre por los hematíes (glóbulos rojos).

**Etapa 4.** Según pasa la sangre a través de los capilares, proporciona oxígeno (y nutrientes como la glucosa) a los tejidos y células corporales y recoge los productos de desecho de la respiración celular: dióxido de carbono y agua.

**Etapa 5.** En el interior de las células corporales, la glucosa y el oxígeno forman parte de una compleja serie de reacciones, que proporcionan energía a las células. Durante esta respiración celular la glucosa es convertida en dióxido de carbono y agua.

**Etapa 6.** El dióxido de carbono es devuelto en la sangre al corazón y luego a los pulmones, donde se expande por los alvéolos y es expulsado del organismo con la respiración.



**Figura 4. Esquema de las etapas de la respiración**

**Fuente: Enciclopedia Larousse**

### 2.1.2. Generación de los Sonidos por Cavidades Resonantes y Física del Sistema Respiratorio

Por medio de la auscultación pulmonar en la que el médico coloca el diafragma, membrana del estetoscopio mientras el paciente inspira y expira alternadamente, se pueden escuchar los movimientos respiratorios en personas sanas y en enfermas del pulmón, porque se producen ruidos apreciables. Se hace necesario auscultar de forma metódica el tórax tanto en su parte posterior, anterior y las parte laterales. Lo ideal para una buena auscultación pulmonar es escuchar sucesivamente los sonidos en lugares homólogos de ambos lados, ya que esto permite una comparación, adecuada para descubrir lesiones unilaterales.

Es preciso tener en cuenta que cuando se ausculta en varones se debe cuidar que el roce con el vello torácico no cause confusiones, lo cual haría mayor la probabilidad de un diagnóstico errado.

La generación del sonido (el término sonido se utiliza para definir un cambio de presión que se propaga en un medio elástico) respiratorio se debe a la circulación del aire en los pulmones. El flujo de aire a través de las vías aéreas causa perturbaciones, que originan las vibraciones que percibimos como ruidos. Las alteraciones se producen en las zonas donde la velocidad del aire es mayor y en aquellas con condiciones geométricas que dificultan un flujo laminar.

Los sonidos generados en los pulmones y en las vías aéreas por la respiración normal de un individuo sano, se diferencian según la localización donde son auscultados.

Los mecanismos de producción de estos ruidos han sido motivo de debate ya que se ha apoyado más en opiniones personales que en observaciones experimentales concretas. Los sonidos normales se encuentran presentes en toda el área torácica. Al auscultar los pulmones se pueden encontrar sonidos respiratorios normales, sonidos disminuidos o ausentes, o sonidos anormales como se muestra en la siguiente clasificación.

Sonidos respiratorios:

**A. Sonidos normales**

Sonido pulmonar (vesicular)

Sonido bronquial

**B. Sonidos normales aumentados**

Respiración ruda (broncopulmonar)

Respiración Traqueal

**C. Sonidos normales disminuidos**

En referencia a:

Ausencia más o menos generalizada (derrame pleural, neumotórax).

Ausencia localizada (enfisema, atelectasia con bronquio ocluido).

**D. Sonidos anormales continuos**

Roncus (roncancias)

Sibilancias

**E. Sonidos anormales discontinuos**

Estertores finos regulares (crepitancias)

Estertores finos irregulares

Estertores gruesos

Frote pleural

Crepitación mediastinal

## F. Sonidos apreciados a distancia

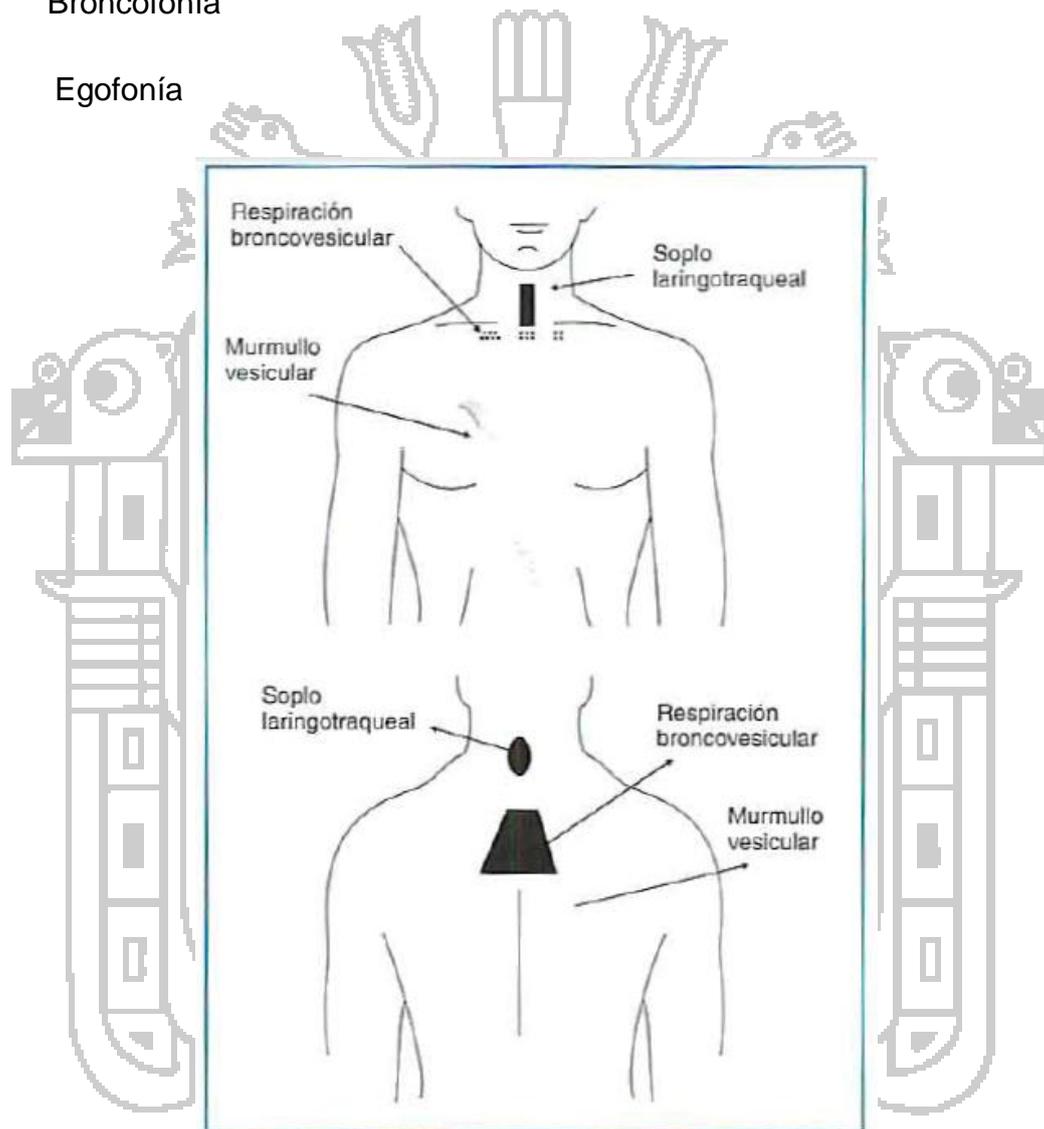
Estridor

Sibilancias

## G. Auscultación de la voz

Broncofonía

Egofonía



**Figura 5. Hallazgos percutóricos normales en las distintas áreas topográficas del tórax.**

**Fuente: Semiología Médica Fisiopatología y Propedéutica**

Algunas características generales de los sonidos de nuestro interés:

### 2.1.2.1. Sonidos Respiratorios Normales

Existen cuatro tipos de sonidos respiratorios, estos pueden ser oídos sobre el pecho durante la respiración normal, en esta clasificación se tiene: sonidos de la respiración pulmonar, esta clase de sonidos se asemejan a un suspiro o a un susurro tranquilo y son escuchados sobre la mayor parte del pecho y con un rango de frecuencia que va de 100 a 1000 Hz. La fase de espiración predomina sobre la inspiración en una relación de 3 a 1, en esta clase de sonidos la fase de espiración es mucho más suave (se percibe usualmente sólo en su primera mitad) y es a menudo inaudible. La respiración Bronquial es producida por el flujo de aire en los bronquios centrales, una de las características de este sonido es de tonalidad fuerte. En contraste con los sonidos de la respiración pulmonar, la espiración es más pronunciada que la inspiración, y hay generalmente una pausa entre las dos fases.

Existen también los sonidos de la respiración Broncopulmonar, como su nombre lo indica son sonidos intermedios entre los sonidos vesiculares y los sonidos bronquiales, de esta forma la inspiración y espiración tienen la misma calidad y duración. Las dos fases son continuas o tienen una separación muy pequeña. Los sonidos broncovesiculares, son normalmente audibles en la parte anterior, cerca del primer y segundo espacio intercostal, y en la parte posterior entre las dos escápulas. En ambas localizaciones los sonidos son más evidentes en la parte derecha que en la izquierda.

Los sonidos de la respiración traqueal, son muy fuertes y agudos, tienen un rango de frecuencia de 100 a 3000 Hz. Tienen componentes iguales en la inspiración y espiración con una pausa definida entre las dos fases. Estos sonidos son escuchados solo sobre la porción extratorácica de la tráquea y no son habitualmente escuchados en la rutina de auscultación.

Los sonidos disminuidos o ausentes son aquellos que no se perciben o se escuchan con un volumen menor, reflejando una disminución en el flujo de aire en alguna zona, una sobre inflación de una porción de los pulmones como en el enfisema, presencia de aire o de líquido alrededor de los pulmones o, incluso pueden reflejar un incremento en el grosor de la pared torácica.

### 2.1.2.2. Sonidos Anormales o Adventicios

Se denominan así a aquellos sonidos que no son propios al proceso normal de respiración y por lo tanto indican enfermedad pulmonar sin que su diagnóstico implique la presencia de un proceso patológico específico.

Los sonidos adventicios son separados en dos categorías principales denominados continuos y discontinuos.

#### 2.1.2.2.1. Sonidos Continuos

Esta clase de sonido indica que tiene una duración de 250ms o más y en el espectro se muestran como picos repentinos separados del sonido respiratorio, y pueden no estar durante el ciclo completo de la respiración. Se pueden auscultar sobre cualquier sitio del tórax. Esta clase de sonidos tienen un carácter musical que de acuerdo a su tono (característica de un sonido que da una percepción definida de frecuencia), se clasifican en sibilancias y roncus, las de tono agudo o alto y de tono bajo o grave respectivamente.

- **Sibilancias:** Esta clase de sonidos respiratorios adventicios emite un silbido, que se produce en el pecho durante la respiración que ocurre por el paso del aire por los conductos respiratorios obstruidos. Estos sonidos son propios del asma, bronquitis, bronquiolitos o edema pulmonar (líquido en los alvéolos pulmonares), así como cuando un cuerpo extraño ingresa en las vías respiratorias.
- **Roncus:** Se produce por la presencia de una secreción abundante en los conductos bronquiales o por un engrosamiento de la pared de los mismos debido a la inflamación de ellos. Este tipo de sonidos respiratorios anormales son característicos de la bronquitis.

La producción de roncus o sibilancias no depende del ancho original de las vías aéreas donde se producen o de su longitud, sino de la relación que exista entre el tamaño de la abertura donde se produce el ruido, la velocidad del aire que lo produce y las propiedades elásticas de las estructura. Un cuerpo extraño o un tumor comprimiendo la tráquea por ejemplo pueden producir roncus o sibilancias según sea la velocidad con que circule el aire.

Los métodos clásicos de auscultación, los cuales se emplean en la práctica médica, describían las sibilancias en términos de todo o nada. Mediante los métodos de análisis por computador es posible realizar medidas cuantitativas de las sibilancias.

El criterio absoluto para afirmar o no la presencia de roncus y sibilancias, es su auscultación. La exploración física por medio de la auscultación, permiten una calificación de los ruidos anormales como roncus, sibilancias, caracterizando de forma subjetiva el predominio de tonos altos (sibilancias) o de tonos bajos (roncus).

#### **2.1.2.2.2. Sonidos Discontinuos**

Se caracterizan porque son de corta duración ( $\leq 10\text{ms}$ ), y se presentan como una serie de sonidos explosivos que al momento de la auscultación se escucha como si se unieran los unos con los otros sin lograrse del todo. Según sus características acústicas se pueden dividir en estertores gruesos o finos.

Estertores finos (se parecen al ruido que se produce al frotar el cabello), estos a su vez también se clasifican en estertores finos de aparición regular y constante al final de la inspiración y los estertores finos de aparición irregular, al comienzo de la inspiración o durante la espiración.

Los estertores finos de aparición regular o llamados también crepitancias se auscultan con regularidad al final de la inspiración como un ruido similar al que se produce al desprender el velcro del manguito de algunos tensiómetros y corresponden al momento en que se abren las vías aéreas periféricas cerradas durante la espiración anterior. Estos sonidos son característicos de la asbestosis, sarcoidosis, de la fibrosis pulmonar y el edema pulmonar, además pueden auscultarse en la fase central de la neumonía y el infarto pulmonar.

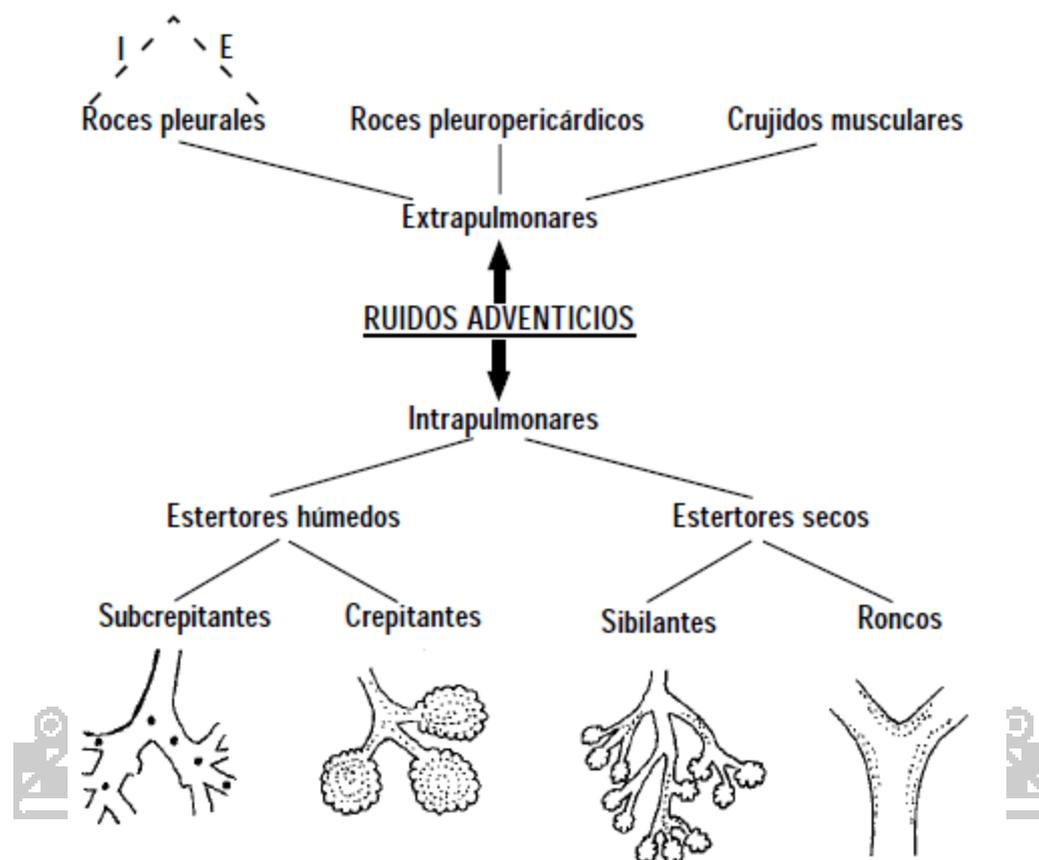
Los estertores finos de aparición irregular se auscultan en cualquier momento del ciclo respiratorio ya sea al principio o al final de la inspiración o espiración, se creen que son producidos por secreciones endobronquiales o cambios inflamatorios en los bronquios de pequeño calibre. Este tipo de sonidos se pueden escuchar en la bronquitis aguda, crónica y el edema pulmonar.

Los estertores gruesos son de mayor duración y mayor intensidad que los estertores finos por lo cual en algunas ocasiones pueden ser audibles a distancia. Son producidos por las burbujas que forman el aire con las secreciones bronquiales por lo cual se cambian o desaparecen con la tos. Estos sonidos se encuentran en el edema pulmonar, la fase evolutiva avanzada de las bronconeumonías, las bronquitis agudas, la exacerbación aguda de las bronquitis crónicas y en general en todos los procesos acompañados de secreciones bronquiales abundantes.

**Tabla I: Tipos de sonidos respiratorios y sus características acústicas principales**

<b>RUIDOS AGREGADOS</b>	<b>ORIGEN</b>	<b>HZ</b>	<b>CARACTERISTICA</b>	<b>PATOLOGIAS</b>
<b>SIBILANCIAS</b>	Bronquios de pequeño calibre "silbido"	600-1000Hz	Tonalidad aguda -Espiración	Asma Bronquiolitis
<b>RONCUS</b>	Bronquios grandes, tráquea	100-3000Hz	Tonalidad grave -Inspiración y espiración	Bronquitis Aguda
<b>ESTERTORES CREPITANTES</b>	Alveolos		Final de la inspiración "como el frote de cabello"	Neumonía Insuficiencia Cardíaca
<b>SUBCREPITANTES</b>	Bronquios pequeños.	600-1000Hz	Inspiración y espiración	Atelectasia
<b>FROTE PLEURAL</b>	Pleura		-Roce de las pleuras "como pisar nieve recién congelada" -Final de la inspiración	Pleuritis

De acuerdo a lo expuesto anteriormente, la auscultación pulmonar en un enfermo de las vías respiratorias debe basarse en la presencia de los sonidos pulmonares tanto normales como anormales, para la determinación de un diagnóstico médico que favorezca a la mejoría de sus pacientes.



**Figura 6. Ruidos adventicios**

**Fuente: Propedéutica Clínica y Semiología Médica**

### 2.1.3. Los Sonidos Respiratorios y su Uso Como Herramienta Diagnóstica

La auscultación realizada con el estetoscopio, permite diagnosticar enfermedades como bronquitis, neumonía, síntomas del edema pulmonar, de la enfermedad pulmonar obstructiva crónica, asma, entre otras; ya que este procedimiento proporciona información de las posibles anomalías como obstrucción del flujo aéreo, alteraciones y síntomas de enfermedades respiratorias; pero el diagnóstico acertado de dichas anomalías se debe a una buena interpretación de los sonidos escuchados, lo cual hace que este dictamen sea muy subjetivo y que dependa del oído, la experiencia y la habilidad del profesional en medicina para diferenciar unos sonidos de otros. En los últimos años, la tecnología ha contribuido con el hecho de brindar herramientas de diagnóstico más acertadas, reforzando y enriqueciendo el dictamen médico, de esto surgen los métodos de análisis computarizado de

sonidos respiratorios como una alternativa novedosa que va más allá de la auscultación simple. Con el análisis de los sonidos respiratorios se pueden medir los cambios en ellos, ya que se hacen grabaciones permanentes y se producen gráficas que ayudan con el diagnóstico para el tratamiento de las enfermedades pulmonares, de esta forma es fácil determinar los cambios de intensidad del sonido, lo que posibilita la identificación de obstrucciones leves que no se obtienen con la auscultación simple, además de estar reduciendo la subjetividad del diagnóstico por parte del médico.

El análisis de los sonidos del pulmón permite ser usado en pacientes de cualquier edad, ya que no requiere de un máximo esfuerzo y se presenta como una promesa en investigaciones sobre patologías de las vías aéreas, así como en pacientes con apnea del sueño obstructiva y con estrechez en la tráquea.

Muchos de los conocimientos ganados recientemente han sido el resultado de una amplia variedad de técnicas de adquisición de datos, procesamiento y análisis de señales, el uso de técnicas modernas de procesamiento de señales digitales, el análisis en la forma de onda de los sonidos del pulmón por computadora, llegará a estabilizar las técnicas de investigación en este tema.

Los sonidos respiratorios contienen información fisiológica y patológica del pulmón y de las vías aéreas. El espectro de frecuencia y la amplitud de los sonidos traqueales o sonidos pulmonares sin componentes adventicios, pueden reflejar la dimensión y cambios patológicos en el tejido pulmonar, los sonidos respiratorios grabados sobre el pulmón tiene su banda de frecuencia fundamental de 200-250 Hz. Uno de los inconvenientes con esta banda de frecuencia es que contiene componentes de los músculos respiratorios y del corazón.

Cuando los sonidos son grabados sobre la tráquea la energía fundamental se extiende por encima de 850-1000 Hz; es importante destacar que los sonidos de la respiración son dependientes de la perturbación inducida por el flujo de aire. Los sonidos bronquiales tienen componentes de frecuencia hasta de 600-1000 Hz, cuando son registrados en la parte posterior del pecho.

De esta forma se determina que la banda de frecuencia más comúnmente usada para los sonidos de la respiración es de 60-100 Hz hasta 2kHz cuando los sonidos son grabados en el pecho y de 60-100 Hz hasta 4kHz cuando estos son obtenidos sobre la tráquea. La frecuencia estándar para los sonidos adventicios en el pecho, se encuentra de 60-100 hasta 6kHz<sup>1</sup>.

El análisis computarizado del sonido del pulmón es un campo que integra no solo el diagnóstico de enfermedades pulmonares sino que permite caracterizar elementos que sirvan para su tratamiento, ya que la grabación y análisis pueden ser usados no solo para determinar enfermedades respiratorias, sino también para obtener la respuesta de broncodilatadores y broncoconstrictores así como las variaciones del flujo de aire durante las pruebas bronquiales.

#### 2.1.4. Enfermedades Respiratorias Comunes

Las enfermedades respiratorias son causadas por microbios que están en el ambiente o en las gotitas de saliva o moco, que una persona enferma arroja al toser, hablar o estornudar.

Las más comunes en infantes son:

- **Coriza (gripa o gripe).** Es un proceso inflamatorio de las vías respiratorias superiores que se manifiesta casi siempre por secreción nasal, estornudos y tos. Suele ser causada por diversos virus, que ocasionan otitis media, sinusitis u otras complicaciones poco frecuentes; fiebre, malestar general y una fuente identificada de infección son datos comunes.
- **Bronquiolitis.** Enfermedad inflamatoria de la vía aérea periférica, afecta al bronquiolo terminal y respiratorio, es la causa más frecuente de hospitalización en lactantes y niños en edad preescolar. Se caracteriza por la inflamación de la vía aérea distal con sobre distensión pulmonar y signos de dificultad respiratoria, inicia con infección de las vías respiratorias superiores (IVRS); después aparece tos, sibilancias, disnea, aleteo nasal, cianosis, tórax en tonel, frecuencia respiratoria

---

<sup>1</sup> Patología Respiratoria, Manual de procedimientos de diagnóstico y control.

aumentada, movilidad disminuida, espiración prolongada y estertores sibilantes, los cuales se convierten en crepitantes y subcrepitantes.

- **Asma.** Enfermedad crónica inflamatoria de las vías aéreas inferiores (IVAI) consecutiva a hiperactividad bronquial que provoca broncoespasmo, edema e hipersecreción manifestándose por tos, disnea, y sibilancias cediendo de forma espontánea o bajo tratamiento. Afecta del 10 al 24 % de la población mundial con predominio masculino en la infancia (entre 2 y 5 años) y femenino de los 10 años en adelante. Su principal etiología (causa) es alérgica, los síntomas son: tos, disnea, sibilancias, a la auscultación dificultad respiratoria, sibilancias, silencio respiratorio y respiración prolongada.
- **Bronquitis.** Es la inflamación de los bronquios, se debe a virus o bacterias, pero el humo del cigarro y la contaminación también pueden ser culpables de este mal. Muchas veces se da después de un catarro que no se curó bien o de una infección respiratoria. Los signos primarios y los síntomas son la disnea y la tos leve persistente que puede o no producir mucosidad. Según progresa la enfermedad, la dificultad para respirar puede limitar la capacidad de la persona para llevar a cabo su actividad diaria.
- **Bronconeumonía.** Es la inflamación de la parte baja de las vías respiratorias (bronquiolos finos y sacos alveolares de los pulmones) debido a una infección; produce serios trastornos pulmonares. Generalmente es causada por microbios (virus y bacterias). Surge como una complicación grave de otras enfermedades respiratorias como resfríos mal cuidados, influenza y bronquitis. Es la primera causa de muerte infantil (principalmente en menores de 1 año). Los síntomas son: fiebre, frecuencia respiratoria agitada, sobre 40 o 60 respiraciones por minuto (dependiendo de la edad), estertores crepitantes y subcrepitantes, y en algunos casos, insuficiencia respiratoria.
- **Neumonía.** Es una infección de uno o de ambos pulmones, de tipo bacteriana, viral o por hongos, que ocasiona una grave inflamación. Esta inflamación produce dificultad para respirar y también dolor. La neumonía por aspiración puede ocurrir cuando una sustancia extraña, como alimento o líquido, entra en los pulmones; es la causa del 70 % de

la mortalidad en los niños menores de 5 años. Se caracteriza por escalofríos, fiebre (39°C o más), tos, expectoración purulenta, dolor pleural, disnea, estertores crepitantes y subcrepitantes en algunos casos insuficiencia respiratoria.

**Tabla II: Patologías más frecuentes**

<b>Alteraciones de la vía aérea</b>	<b>Alteración de la pleura y de la pared torácica</b>
<ul style="list-style-type: none"> <li>• Asma</li> <li>• Agudización de bronquitis crónica o enfisema</li> <li>• Obstrucción de la vía aérea de otro origen</li> </ul>	<ul style="list-style-type: none"> <li>• Tórax inestable</li> <li>• Neumotórax</li> <li>• Derrame pleural</li> </ul>
<b>Afectación del parénquima pulmonar</b>	<b>Enfermedades neuromusculares</b>
<ul style="list-style-type: none"> <li>• Insuficiencia cardíaca congestiva</li> <li>• Neumonía</li> <li>• Aspiración</li> <li>• Síndrome de distrés respiratorio agudo</li> </ul>	<ul style="list-style-type: none"> <li>• Síndrome de Guillain-Barré</li> <li>• Botulismo</li> <li>• Miastenia grave</li> <li>• Poliomielitis</li> <li>• Lesión medular</li> <li>• Ictus</li> <li>• Traumatismo craneoencefálico</li> <li>• Sobredosis de opiáceos, sedantes o hipnóticos</li> </ul>
<b>Afectación vascular</b>	
<ul style="list-style-type: none"> <li>• Embolia pulmonar</li> </ul>	

### 2.1.5. Filtros

Un filtro es un circuito selectivo en frecuencia, los filtros se diseñan para dejar pasar algunas frecuencias y rechazar otras. Hay numerosas formas para utilizar los circuitos de filtros. Los filtros sencillos creados con el uso de resistores y capacitores o con inductores y capacitores se denominan filtros pasivos porque utilizan componentes pasivos que no amplifican. Un tipo especial son los filtros activos que emplean redes RC con realimentación en circuitos con amplificadores operacionales que son caracterizados por ofrecer exactitud y estabilidad; sin embargo, es difícil diseñar un selector de frecuencia ideal, para esto utilizamos las aproximaciones de filtros pues entre más sea la aproximación que se desea, más elevado es el orden del filtro.

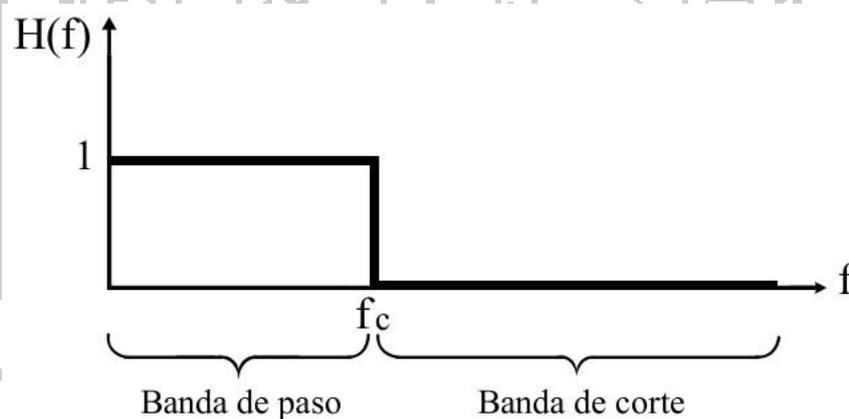
Los selectores de frecuencia son filtros que permiten el paso de las frecuencias dentro de ciertas bandas, llamadas bandas de paso y bloquean las frecuencias en otras bandas, denominadas bandas de corte. Idealmente en las bandas de paso debería haber transmisión sin distorsión, y en las bandas de corte la ganancia debería ser nula. Existen cuatro tipos básicos de selectores de frecuencia: pasa bajas, pasa altas, pasa banda y rechaza banda, que mencionaremos en las secciones siguientes.

### 2.1.5.1. Filtro Pasa Baja

La respuesta pasa bajas se caracteriza por una frecuencia  $f_c$ , que se llama la frecuencia de corte, tal que  $|H| = 1$  para  $f < f_c$  y  $|H| = 0$  para  $f > f_c$  lo que indica que las señales de entrada con frecuencia menor que  $f_c$  pasan a través del filtro sin cambiar su amplitud, mientras que las señales con  $f > f_c$  sufren atenuación total. Una aplicación común de los filtros pasa bajas es la remoción de una señal de ruido de alta frecuencia.

Su comportamiento será:

$$|H(f)| = \begin{cases} 1 & \text{si } f < f_c \\ 0 & \text{si } f > f_c \end{cases}$$



**Figura 7. Diagramas de un filtro pasa baja ideal**



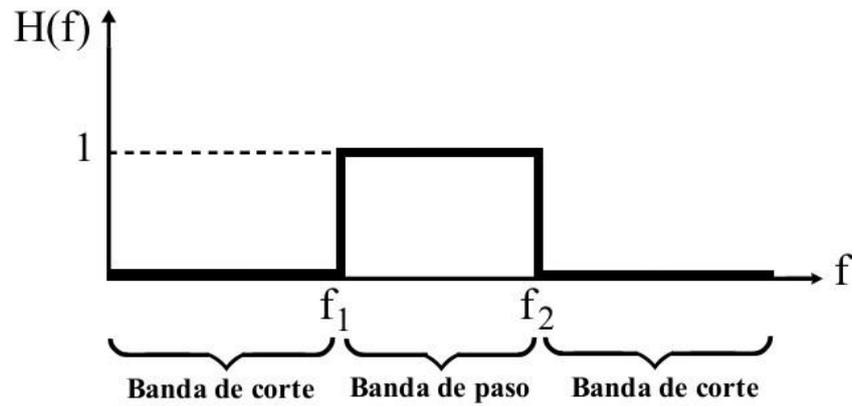


Figura 9. Diagrama de un filtro pasa banda ideal

#### 2.1.5.4. Filtro rechaza Banda

La respuesta rechaza banda es complementaria a la respuesta pasa banda porque bloquea los componentes de la frecuencia dentro de la banda de rechazo  $f_1 < f < f_2$ , al tiempo que pasan todas las demás. Cuando la banda de rechazo es suficientemente estrecha, la respuesta se llama notch. Una aplicación de los filtros notch es la eliminación de la frecuencia de la línea de 60 Hz. en los equipos médicos.

$$H(\omega) = \begin{cases} 0 & \text{si } f_1 < f < f_2 \\ 1 & \text{si } f < f_1; f > f_2 \end{cases}$$

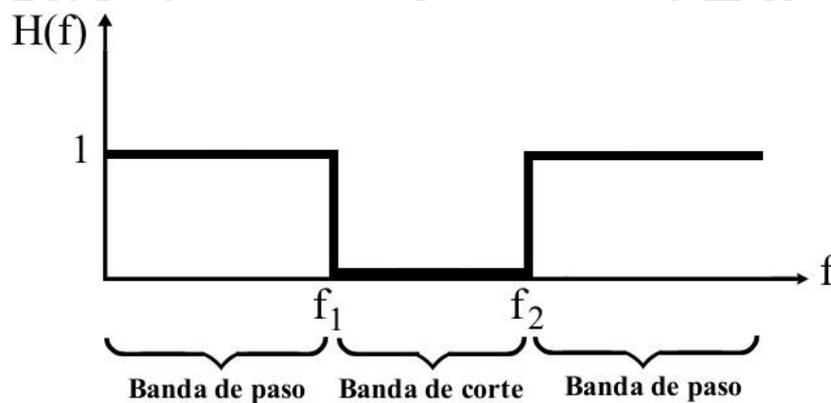
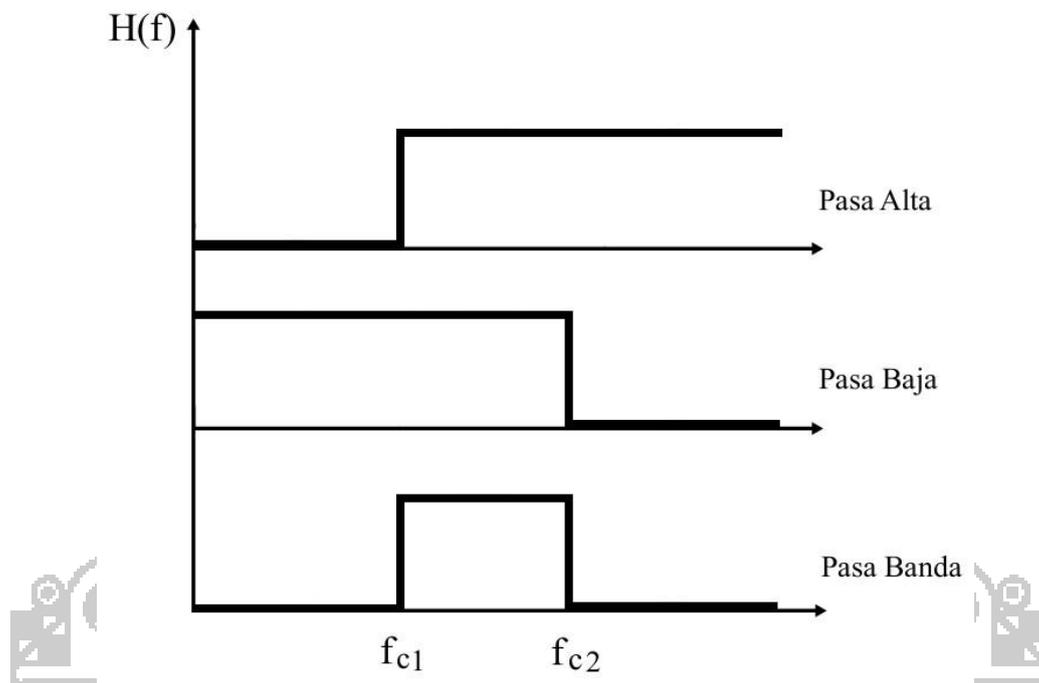


Figura 10. Diagrama de un filtro rechaza banda ideal

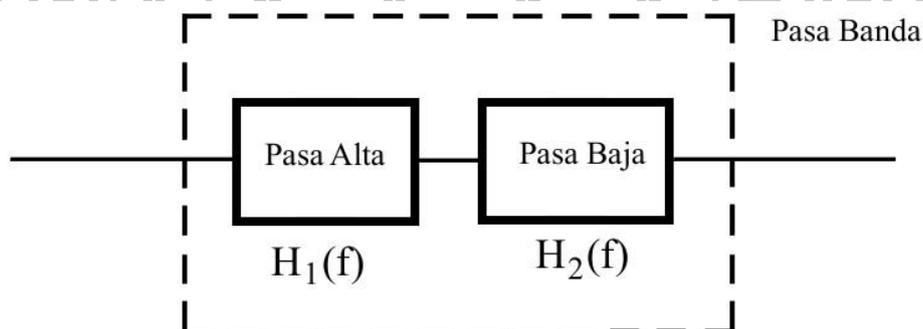
De estos cuatro filtros los dos primeros son imprescindibles ya que los otros dos se pueden obtener a partir de esos dos primeros. Así, por ejemplo, si

colocamos en serie un filtro pasa alta con  $f_{c1}$  seguido de un filtro paso de baja con  $f_{c2}$ , siendo  $f_{c2} > f_{c1}$  obtenemos un filtro pasa banda.



**Figura 11. Resultado de un filtro pasa banda**

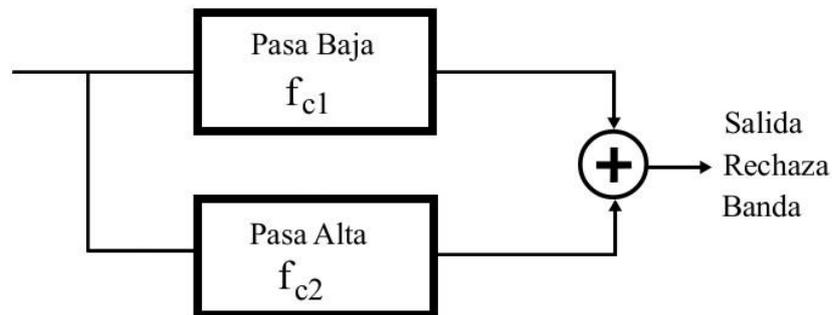
Vemos que se obtiene un filtro pasa banda, que permite el paso de las frecuencias comprendidas entre  $f_{c1}$  y  $f_{c2}$ . La función de transferencia se obtiene multiplicando las funciones de transferencias correspondientes  $H(f) = H_1(f) \cdot H_2(f)$ .



**Figura 12. Diagrama de bloques de filtro pasa banda**

Algo parecido se puede hacer para conseguir un filtro de rechaza banda, aunque ahora no se colocan los dos filtros en cascada. Utilizaremos un filtro

paso de baja con  $f_{c1}$  y un filtro paso de alta con  $f_{c2}$ . Siendo  $f_{c2} > f_{c1}$  y los colocaremos en paralelo con un sumador a la salida de ambos filtros.



**Figura 13. Diagrama de bloques de filtro rechaza banda**

El comportamiento del conjunto se refleja en el siguiente dibujo:



**Figura 14. Resultado de un filtro rechaza banda**

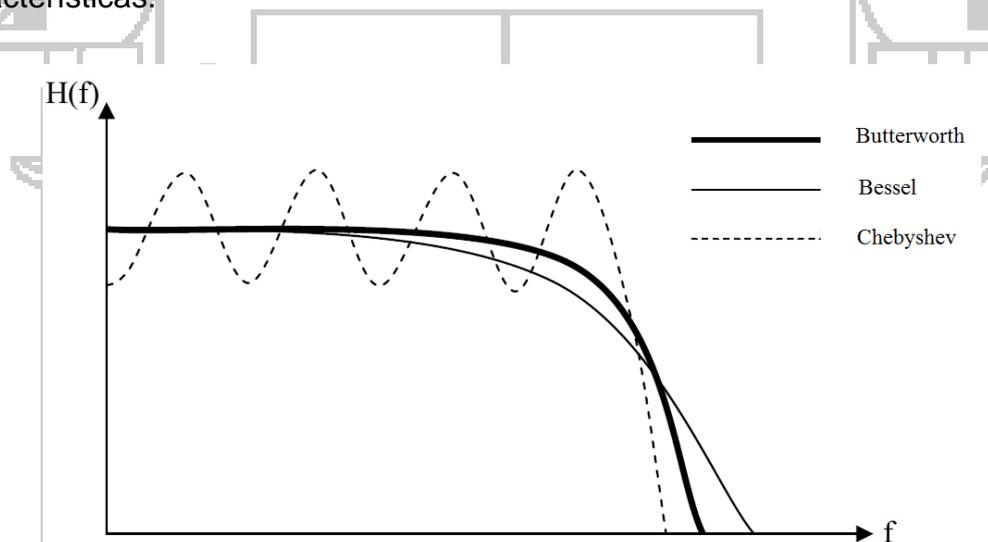
Por lo que hemos indicado más arriba, está claro que sólo es necesario estudiar dos tipos de filtro de respuesta en frecuencia: filtro de paso de baja y filtro de paso de alta.

### 2.1.5.5. Características de la Respuesta de un Filtro

Cada tipo de respuesta (pasa bajos, pasa altos, pasa banda o rechaza banda) puede diseñarse mediante valores de componentes del circuito a fin de tener una característica Butterworth, Chebyshev o Bessel. Cada una de estas características se identifica por la forma de la curva de respuesta y cada una tiene una ventaja en ciertas aplicaciones.

Las características de respuesta Butterworth, Chebyshev o Bessel pueden realizarse con casi todas las configuraciones de circuito para filtros activos mediante la selección apropiada de ciertos valores de componentes.

En la figura 15 se muestra una comparación general de las tres características de respuesta para una curva de respuesta pasa bajos. También es posible diseñar filtros pasa-altos y pasa-bandas que tengan cualquiera de las tres características.



**Figura 15. Gráfica comparativa de tres tipos de características de respuesta de filtros**

En la figura se muestra una primera aproximación Butterworth que corresponde a una función cuya magnitud es monótona, es decir, su magnitud siempre disminuye (o aumenta) cuando la frecuencia aumenta. Una aproximación de Bessel que tiene una banda pasante plana y una banda eliminada monótonica parecida a la de la aproximación de Butterworth. Para un filtro del mismo orden, sin embargo, la caída en la zona de atenuación es

mucho menor. También se tiene la aproximación tipo Chebyshev que presenta un rizo en la banda de paso y que es monótona en la banda de rechazo.

#### 2.1.5.5. Introducción al Filtro Butterworth

En diversas aplicaciones de los filtros se necesita que la ganancia en lazo cerrado se aproxime lo más posible a 1 dentro de la banda de paso, está a proximidad es cuando el amplificador operacional se utiliza como amplificador de ganancia unitaria. Para este tipo de aplicación lo mejor es el filtro Butterworth. A este tipo de filtro también se le conoce como filtro máximamente plano o plano-plano. En el presente trabajo todos los filtros utilizados serán del tipo Butterworth. En las figuras 16 y 17 se muestra la respuesta a la frecuencia ideal (línea continua) y la respuesta a la frecuencia real (líneas punteadas) de tres tipos de filtros Butterworth. Conforme las atenuaciones se van volviendo más pronunciadas, podemos ver que se aproximan más al filtro ideal.

Los filtros Butterworth no se diseñan para mantener un ángulo de fase constante en la frecuencia de corte, la respuesta decae a un ritmo aproximado de  $20n$  dB por década, donde  $n$  es el orden del filtro. Para el filtro pasa bajas básico de  $-20$  dB/década tiene un ángulo de fase de  $-45^\circ$  en la frecuencia  $f_c$ . El filtro Butterworth de  $-40$  dB/década tiene un ángulo de fase de  $-90^\circ$  en el valor  $f_c$  y el filtro de  $-60$  dB/década tiene un ángulo de fase de  $-135^\circ$  en  $f_c$ . Por lo tanto, por cada aumento de  $-20$  dB/década, el ángulo de fase aumenta en  $-45^\circ$  en el valor  $f_c$ .

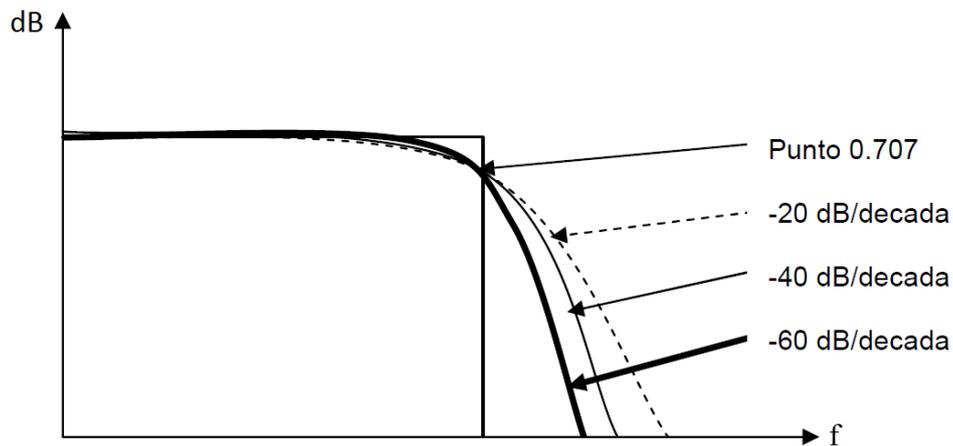


Figura 16. Comparación de la respuesta a la frecuencia de tres filtros pasa altas

Butterworth

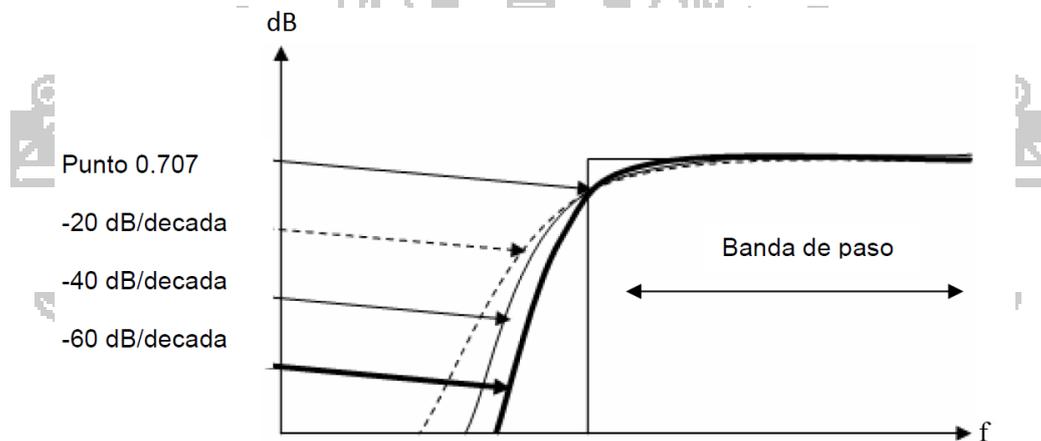


Figura 17. Comparación de la respuesta a la frecuencia de tres filtros pasa altas

Butterworth

## 2.1.6. Red de Telefonía IP

### 2.1.6.1. Introducción

En 1980, la Organización Internacional de estándares crea el modelo de referencia OSI que fragmenta la comunicación entre máquinas en 7 capas distintas.

Cada capa se ocupa sólo de hablar con su correspondiente capa situada en la otra máquina.

A su vez cada capa proporciona servicios a la que está por encima de ella y solicita servicios de la capa directamente por debajo.

7	Aplicación	NFS
6	Presentación	XDR
5	Sesión	RPC
4	Transporte	TCP, UDP
3	Red	IP, ICMP, Protocolos de enrutamiento
2	Enlace	ARP, RARP
1	Física	No especificados

**Figura 18. Modelo de referencia OSI**

Debido a la necesidad de transmitir las tramas en un tiempo lo más inmediato posible, se recurre al protocolo UDP para el transporte de voz.

Sin embargo, no es suficiente con el uso de UDP, la IETF desarrolla el RTP como protocolo para transmitir tráfico sensible al retraso en las redes basadas en conmutación de paquetes. Este nuevo protocolo contiene una información esencial para determinar los tiempos de llegada y el orden de los paquetes.

Además consta de una parte de datos y una parte de control (protocolo RTCP).

La voz sobre IP se encapsula dentro de RTP, que a su vez se encapsula dentro de UDP, por tanto la cabecera del paquete VoIP será RTP/UDP/IP.



**Figura 19. Encapsulamiento VoIP**

## 2.1.6.2. Protocolos Relacionados con VoIP

### 2.1.6.2.1. Protocolo IP

El protocolo de Internet o protocolo IP, se denomina sin conexión y pertenece a la capa 3. No proporciona mecanismo de actuación para dotar al sistema de fiabilidad, control de flujo, secuenciación o reconocimiento.

Sin embargo se puede ejecutar IP a través de cualquier medio dada su posición en el modelo OSI. Es posible escribir una aplicación y tenerla para un conjunto de medios en cualquier sitio.

Existen distintos tipos de paquetes IP:

- **Unidifusión:** Sólo identifica una dirección de destino específica. Permite que dos estaciones se comuniquen independientemente de su ubicación.
- **Difusión:** Paquetes enviados a todos los usuarios de una subred local. Pueden atravesar puentes y switches, pero no routers. Se usan para la comunicación de toda la subred.
- **Multidifusión:** Usan una gama especial de direcciones que permiten la comunicación entre usuarios de redes distintas. Permiten aplicaciones que tienen un transmisor y múltiples receptores, como por ejemplo en una videoconferencia.

### 2.1.6.2.2. Protocolo UDP

En la capa de transporte se ha optado por el uso del protocolo UDP entre las posibles opciones para el transporte de IP.

Ha sido elegido en detrimento de TCP por la necesidad de una alta velocidad en la transmisión de los datos. Sin embargo, no proporciona seguridad en la entrega de los paquetes.

Aunque TCP es fiable y garantiza la entrega, es demasiado lento para transmitir señales de audio que deben llegar con un retardo mínimo.

Se trata de un protocolo más sencillo que TCP.

Se define como protocolo sin conexión y contiene una cabecera más pequeña, lo que produce un menor coste temporal en la transferencia.

Los campos UDP son:

- puerto origen
- puerto destino
- longitud (longitud de cabecera + longitud de datos)
- suma de comprobación (opcional)

No hay retransmisión de paquetes, por tanto existe la posibilidad de que se puedan perder algunos por el camino.

Se usa en VoIP para transportar el tráfico de voz real, puesto que en VoIP controlar la latencia es más importante que asegurar la entrega fiable de cada paquete.

#### **2.1.6.2.3. Protocolo RTP (Real-time Transport Protocol)**

RTP es un protocolo de transporte de audio usado por UDP para el transporte de información en tiempo real en redes de conmutación de paquetes.

Proporciona a las estaciones receptoras un número de secuencia, que permite ver si los paquetes llegan en orden, y una marca de temporización, que determina el tiempo de llegada entre paquetes.

Contiene una parte de datos y una de control.

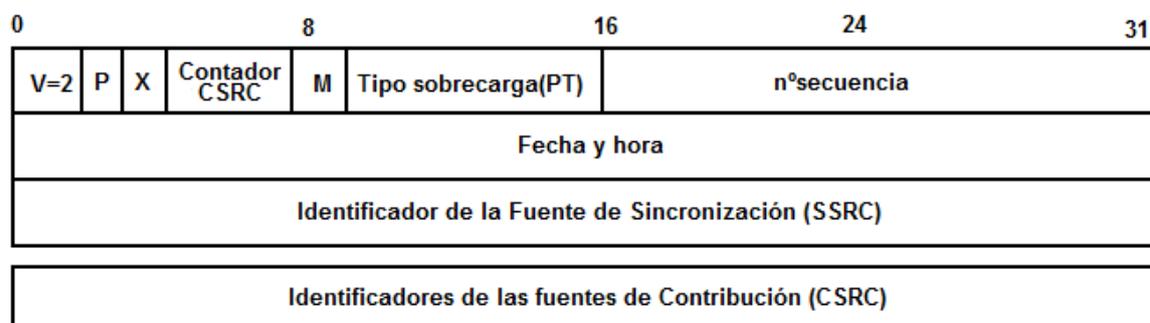
La parte de datos es el protocolo TCP propiamente dicho, da soporte a aplicaciones con propiedades de tiempo real gestionando la reconstrucción de la temporización, la detección de pérdidas y la identificación de contenidos.

La parte de control RTP está gestionada por el protocolo RTCP.

Los paquetes VoIP finales contendrán una o más muestras de códec de voz o tramas encapsuladas en cabeceras IP/UDP/RTP.

UDP proporciona los servicios de entramado y multiplexación de la aplicación para VoIP (a través de los puertos UDP). Y RTP proporciona los servicios adicionales que se necesitan para el transporte de datos en tiempo real.

La cabecera RTP tendrá la siguiente forma:

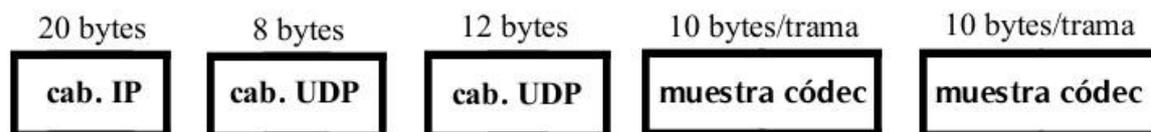


**Figura 20. Cabecera RTP**

Los campos que conforman la cabecera son los siguientes:

- V: versión de RTP con valor 2, compatible con RFC 1889.
- P: Padding, existencia de bits de relleno.
- X: Extensión, indica si hay extensión de cabecera de longitud variable.
- CC: Contador CSRC (CC), número de campos de cabecera CSRC que sigue a la cabecera, 12 bytes.
- M: Marker, identifica el comienzo de un paquete de información speech burst para VoIP.
- PT: Tipo de sobrecarga, identifica los códec de audio.
- n° secuencia: Permite detectar paquetes perdidos.
- fecha y hora: Permite corregir el tiempo de temporización de la sobrecarga.
- SSRC: Fuente de envío, identificador de emisor durante sesión.
- CSRC: Fuentes de contribución, todos los SSRC que se mezclan en una conferencia.

El tamaño total de la cabecera RTP podrá variar entre 8, 12 y 20 bytes. Esto supone un transporte ineficaz de mensajes VoIP, puesto que las cabeceras tienen el doble de bits que la información transportada.



**Figura 21. Tamaño de cabeceras IP/UDP/RTP**

Una solución a este problema es comprimir las cabeceras a 2 ó 4 bytes, usando la compresión de cabecera RTP (CRTP)

Los servicios proporcionados son:

- Distinción entre los emisores múltiples en un flujo multidifusión RTP (gracias a SSRC y CSRC).
- Conserva la relación de temporización entre los paquetes. El bit M identifica el comienzo de una conversación entrecortada, mientras que el buffer playout controla las ráfagas de paquetes.
- Posibilita la sincronización de temporización entre los flujos de medios gracias al campo fecha y hora.
- Secuencia los datos para identificar los paquetes perdidos. Para ello usa el número de secuencia, que aunque no son retransmitidos, pueden ser devueltos gracias a los informes de RTCP.
- Identifica los tipos de medios. El tipo de sobrecarga (PT) refleja el códec de audio usado.
- No asegura la calidad de servicio.

#### 2.1.6.2.4. Protocolo RTCP (Real-time Transport Control Protocol)

RTCP es un protocolo que trabaja junto a RTP complementando algunas de sus funciones como:

- Envío de información periódica sobre la calidad de servicio.
- Control de la sesión con el uso del paquete BYE.
- Identificación. Los paquetes incluyen los datos de todos los participantes de la sesión.
- Sincronización entre medios. Provee la información necesaria para que se reproduzcan a la vez el audio y el vídeo.

- Permite administrar informes.
- Es posible el uso de conferencias RTP multimedia.

Es especialmente útil en VoIP punto a punto para proporcionar retroalimentación de calidad de servicio desde el receptor al emisor.

### **2.1.6.3. Protocolos de Control y Señalización de Llamadas**

Los protocolos asociados a VoIP se pueden dividir en dos grupos: los que transportan la ruta de audio, y los de señalización de llamadas y funciones de control.

Los que transportan la ruta de audio contienen información de temporización para asegurar una reproducción de audio consistente en la recepción. Además permiten la retroalimentación del rendimiento de la Calidad de Servicio (QoS) respecto a la red subyacente.

RTP y RTCP son protocolos de transporte de audio.

Aquellos que se encargan de la señalización de llamadas y las funciones de control permiten configurar y cancelar la llamada. Además incluyen las distintas formas de direccionamiento y enrutamiento así como servicios de información adicionales y métodos para trabajar con otros tipos de señalización.

Entre los más comunes en el tratamiento de la voz sobre IP nos encontramos con:

#### **2.1.6.3.1. Protocolo H.323**

Especifica cómo se transporta el tráfico multimedia sobre las redes de paquetes.

Utiliza los estándares existentes.

Es un protocolo bastante complejo creado para permitir la ejecución de aplicaciones multimedia sobre redes de datos “no fiables”.

Las aplicaciones multimedia de vídeo y datos ocupan la mayor parte del protocolo. Dichas aplicaciones requieren un gran trabajo para ser escalables con H.323, necesitan especificaciones separadas.

#### **2.1.6.3.2. Protocolo SIP (Session Initiation Protocol)**

Protocolo de control de la capa de aplicación.

Se usa para crear, modificar y terminar sesiones con uno o más participantes.

Las sesiones multimedia incluyen audio vídeo y datos. Además, permite que los participantes sean invitados a una conferencia improvisada.

Permite sesiones multimedia de multidifusión y unidifusión.

Está poco implementado a pesar del interés de los fabricantes y clientes.

#### **2.1.6.3.3. Protocolos SGCP y MGCP**

Permiten que un dispositivo central (agente de llamadas, MGC) pueda controlar los puntos finales (MG).

Usualmente se les conoce como xGCP.

Se pueden desarrollar aplicaciones mediante el uso de API basadas en estándares que comunican con los MGC y ofrecen funcionalidad adicional.

#### **2.1.6.3.4. Protocolo IAX2 (Inter Asterisk Exchange)**

Protocolo para la comunicación entre PBXs Asterisk en reemplazo de IAX.

Incorpora mejoras al protocolo original.

Permite minimizar el ancho de banda usado en las transmisiones de control y multimedia de VoIP.

Evita problemas de NAT (Network Address Translation).

Proporciona un soporte para transmitir los planes de marcación.

### 2.1.6.3.5. Protocolo DUNDi (Distributed Universal Number Discovery)

Protocolo de enrutamiento por el cual se puede compartir el Dialplan entre dos sistemas Asterisk.

DUNDi permite a los nodos enlazados compartir información sobre sus respectivos dialplans entre ellos. No es un transporte de llamadas, sino que proporciona información de direccionamiento de números. Por poner un ejemplo, nos permite preguntar a tu nodo vecino si sabe cómo contactar con una extensión telefónica concreta o un cliente de VoIP. Una especie de agenda de teléfonos P2P.

Para que DUNDi funcione, cada cliente DUNDi tiene que conocer por lo menos otro cliente DUNDi. Como estos clientes DUNDi pertenecen a la misma red, se les llama nodos. Además, al menos uno de los nodos alcanzables debe conocer cómo llegar a la extensión consultada.

### 2.1.6.4. Calidad

#### 2.1.6.4.1. Retraso, latencia

Se denomina retraso o latencia en VoIP al tiempo que tarda la voz en salir de la boca del que está hablando y el tiempo en llegar al oído del que está escuchando.

La recomendación G.114 de la ITU-T define que para que exista una buena calidad de voz no debe darse un retraso mayor de 150 ms en un sentido y de extremo a extremo. Por lo tanto, para que exista una buena calidad de voz la suma de los retrasos tiene que ser menor a 150 ms.

Existen tres principales causas de retraso en las redes telefónicas:

#### **Retraso de propagación:**

El retraso de propagación depende de la velocidad de la luz en un medio determinado, como fibra óptica o cable de cobre.

La velocidad de la luz en el vacío es de 300.000 Km/s, sin embargo, tanto en cobre como en fibra óptica puede alcanzar unos 200.000 Km/s.

Suponiendo que exista una red que viaje alrededor del mundo (21.000 Km), este retraso de sentido único sería de 70 ms, y por lo tanto prácticamente imperceptible para el ser humano.

### **Retraso de manejo:**

El retraso de manejo está provocado por los dispositivos durante el envío de la trama.

El retraso de manejo atiende a distintas causas, empaquetado, compresión y switching de paquetes.

Es un gran problema en entornos de paquetes.

El procesador digital de señal (DSP) genera una muestra cada 10 ms si se usa el códec G.729.

### **Retraso en gestión de colas o serialización:**

Se define el retraso de serialización como el tiempo que se tarda en colocar un bit en una interfaz. O también como el tiempo que se necesita para mover un paquete hasta la cola de salida.

En la gestión de colas, el retraso puede referirse al envío de más paquetes que puede manejar la interfaz en un momento dado.

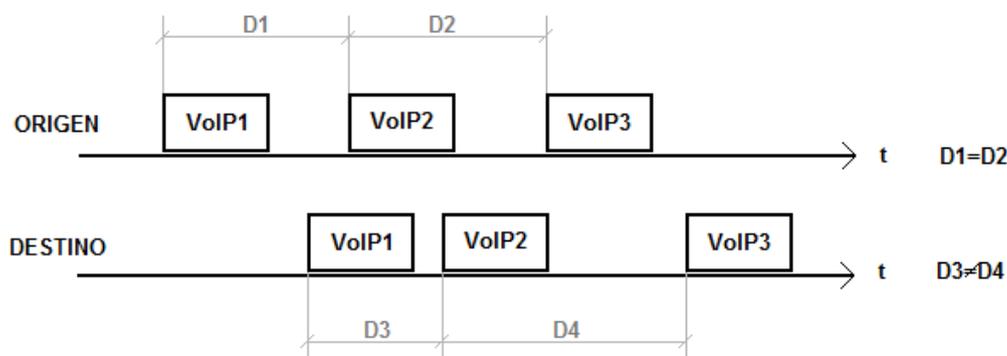
#### **2.1.6.4 .2. Variación de Retardo (jitter)**

Es la variación del tiempo de llegada de los paquetes al receptor. Depende del retraso sufrido en la red, por lo tanto es variable.

La calidad de voz es totalmente dependiente de la temporización con la que las muestras de sonido se ejecutan en el oyente.

Es un retraso exclusivo de las redes de conmutación de paquetes.

Si la temporización entre tramas en el emisor es distinta que en el receptor, éste no oirá una representación exacta del sonido original.



**Figura 22. Variación de retardo**

Existen distintas estrategias para minimizar y gestionar la fluctuación de fase, como priorizar el tráfico de voz por delante de otros tráficos de datos, disminuir el tamaño máximo de paquete para minimizar el retraso por serialización variable, evitar rutas de transmisión paralelas con distintos elementos de retraso, o aumentar la longitud de los búferes de reproducción para alojar la alta variación de retraso.

#### 2.1.6.4 .3. Compresión de Voz

Existen dos métodos de codificación de la voz por forma de ondas: PCM y ADPCM.

##### **PCM de 64 Kbps:**

Contiene en forma de leyes la forma de codificación:

- ley  $\mu$ : Proporciona una mejora de rendimiento en relación señal/ruido. Es usada en América del Norte.  
Para llamadas de larga distancia que requieren un cambio entre la ley  $\mu$  y la ley  $\alpha$ , es responsabilidad del país de ley  $\mu$ .
- ley  $\alpha$ : Es usada normalmente en Europa.

Ambas se diferencian en detalles de compresión relativamente menores. Usan compresión logarítmica para alcanzar de 12 a 13 bits de calidad PCM lineal en 8 bits.

#### **ADPCM:**

La codificación Adaptive Differential Pulse Code Modulation (ADPCM) realiza una codificación usando muestras de 4 bits.

Tiene una velocidad de transmisión de 32 Kbps.

Mide las diferencias de amplitud y la velocidad de cambio de la amplitud.

Usa una predicción lineal rudimentaria.

Además de estos dos tipos, existen nuevas técnicas desarrolladas en los últimos 15 ó 20 años que realizan una codificación basada en la vibración y la modulación, como son LPC, CELP y MP-MLP.

Existen una serie de normas que usan estos sistemas, como son:

- G.711: PCM de 64 Kbps
- G.726: ADPCM de 40, 32, 24 y 16 Kbps
- G.728: CELP de 16 Kbps
- G.729: CELP de 8 Kbps
- G.723.1: CELP de 5'3 Kbps y MP-MLQ de 6'3 Kbps

#### **2.1.6.4 .4. Pérdida de Paquetes**

En las redes de datos una pérdida de paquetes es común y esperada, además de una medida para conocer las condiciones de la red y poder reducir el número de paquetes que se están enviando.

Sin embargo, una pérdida de paquetes de voz es algo que merece la pena mejorar. Es importante construir una red que transporte con éxito la voz de manera fiable y oportuna.

Es necesario crear un mecanismo para hacer que la voz sea resistente a la pérdida periódica de paquetes.

## 2.1.7. Asterisk

### 2.1.7.1. Introducción

Asterisk, es una implementación "open source" de una centralita telefónica (PBX: Private Branch Exchange). Como cualquier PBX, Asterisk permite a un cierto número de teléfonos conectados a él realizar llamadas entre ellos y conectarse a otros servicios telefónicos, incluido la RTC. Su nombre viene del símbolo '\*'.

Asterisk es editado bajo una doble licencia, por una parte posee una licencia de software libre, GNU Public License (GPL), y por el otro lado posee una licencia comercial, para permitirle ejecutar código cerrado o patentado, tal y como ocurre con el codec G.729 (aunque el codec G.729 puede trabajar tanto con versiones comerciales o libres). Mark Spencer, fundador de la empresa Digium, originariamente creó Asterisk y permanece como su principal mantenedor, aunque siguiendo el método de desarrollo de los proyectos de software libre, existen una docena de programadores que han contribuido con nuevas características, funcionalidades y reportando errores. Originariamente diseñado para el sistema operativo Linux, Asterisk ahora también se ejecuta sobre OpenBSD, FreeBSD, Mac OS X, Sun Solaris y Microsoft Windows, aunque como plataforma nativa, Linux es el sistema operativo mejor soportado.

El software básico de Asterisk incluye bastantes características, previamente sólo disponibles en sistemas PBX propietarios, tales como: buzón de voz, conferencia de llamadas, respuesta interactiva y distribución automática, entre otras. Los usuarios pueden añadir nuevas funcionalidades de varias formas: desarrollando scripts en el lenguaje propio de Asterisk, que posteriormente serán interpretados por éste; añadiendo módulos personalizados escritos en C; o escribiendo AGI (Asterisk Gateway Interface) scripts, en Perl u otros lenguajes.

Para conectar teléfonos tradicionales a un servidor Linux ejecutando Asterisk, o para tener acceso a la RTC, el servidor deber ser equipado con cierto hardware (un simple módem no será suficiente). Digium y otras más venden tarjetas PCI

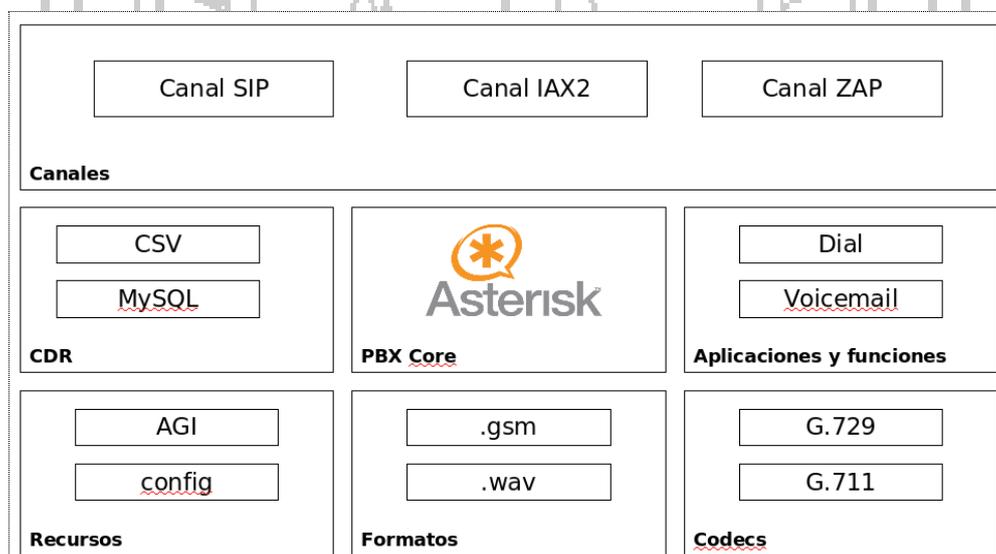
para conectar líneas de teléfonos, líneas T1 o E1, y otros servicios telefónicos analógicos o digitales al servidor.

Puede decirse que, hoy en día, el mayor interés que recibe Asterisk, se debe en parte, al soporte que presenta ante un amplio rango de protocolos de VoIP, incluyendo SIP y H.323. Asterisk, puede interoperar con teléfonos SIP, actuando como un servidor de registro y como Gateway entre los teléfonos IP y la RTC. Los desarrolladores de Asterisk, también han diseñado un nuevo protocolo, IAX, para una reciente comunicación entre servidores Asterisk.

Hasta el término de este proyecto la versión de Asterisk se encuentra en 13.

### 2.1.7.2. Arquitectura de Asterisk

El modelo que se siguió a la hora de crear la arquitectura de Asterisk permite que la aplicación sea altamente escalable y flexible. La arquitectura de Asterisk es modular, lo que permite construir un sistema funcional en base a nuestros requerimientos.



**Figura 23. Arquitectura de Asterisk**

En el centro de la figura 23 de la arquitectura de Asterisk se encuentra el PBX Core, el núcleo de la arquitectura. El núcleo provee al resto del sistema de la funcionalidad básica y posibilita la carga de los distintos módulos, que es lo que dota a Asterisk de su gran funcionalidad.

Un módulo importante es el de aplicaciones y funciones. En él se encuentran las utilidades que más se utilizan a la hora de configurar Asterisk. Por ejemplo, para dotar a Asterisk de servicio de buzón de voz se hace uso de la utilidad VoiceMail, para que Asterisk pueda crear conferencias se hace uso de la utilidad MeetMe, para música en espera MusicOnHold, etc. Existen multitud de utilidades que adicionalmente se pueden personalizar.

Otros módulos, como formatos, permite a Asterisk la gestión de los archivos .gsm, .wav utilizados por lo general para las utilidades de música en espera, buzón de voz, etc. El módulo de codecs que permite a Asterisk codificar/decodificar la información tanto de audio como de video.

En definitiva, mediante esta arquitectura se obtiene un sistema muy escalable y flexible. Escalable porque el funcionamiento independiente de los módulos permite que a la hora de crear uno nuevo solo sea necesario tener en cuenta cómo se van a comunicar con el resto y flexible porque la configuración del sistema se puede hacer, más o menos compleja, a la vez que funcional dependiendo de nuestras necesidades.

## **2.2. MARCO CONCEPTUAL**

### **2.2.1. Sonidos Respiratorios**

El ruido respiratorio se define como el sonido que produce el aire al entrar y salir en el aparato respiratorio. En condiciones normales, si se ausculta sobre las vías aéreas grandes, se oye un sonido profundo y fuerte en el que la espiración resulta más larga que la inspiración llamado “ruido bronquial”; si se ausculta sobre cualquier otro punto del tórax, el sonido es más suave, menos agudo y es fundamentalmente inspiratorio: se denomina “murmullo vesicular”. Podemos clasificar las modificaciones en la auscultación de estos ruidos normales en función del ritmo, la intensidad y el timbre

### **2.2.2. Auscultación Pulmonar**

Como método de exploración clínica, consiste en la exploración auditiva de los sonidos respiratorios (normales o no) que producen los órganos, permite correlacionar esto con las distintas características fisiopatologías de cada

enfermedad. Se efectúa directamente con el oído o el estetoscopio aplicados sobre la piel del tórax. Estos sonidos pueden ser originados por el propio órgano (e.g., contracción cardiaca, borborismos intestinales, etc.), pero sin que este tome su participación inmediata en su producción. Aunque Hipócrates ya conocía no solo el ruido de succión que llevaba su nombre, sino también los ruidos pleuríticos, los cuales comparaba al crujido de un trozo de cuero. El descubrimiento de la auscultación se le atribuye a Laennec (1781–1826) quien también fue el primero en emplear el estetoscopio.

La correcta auscultación respiratoria proporciona signos de gran valor práctico a partir del examen del ruido respiratorio, los ruidos adventicios y la transmisión de la voz.

### **2.2.3. Estetoscopio**

También conocido como fonendoscopio, es un aparato acústico usado en medicina, veterinaria y enfermería para escuchar los sonidos internos del cuerpo humano o animal. Generalmente se usa en la auscultación de los sonidos cardiacos o respiratorios, aunque algunas veces también se utiliza para escuchar ruidos intestinales o soplos por flujos anómalos sanguíneos en arterias y venas. Fue inventado en Francia por el médico Rene Laennec en 1816 debido a la vergüenza que sentía al acercar su oído al pecho de las pacientes.

### **2.2.4. Asterisk**

Asterisk es el más poderoso, flexible y extenso software de telecomunicaciones disponible. Su nombre viene del símbolo asterisco “\*”, que en ambientes UNIX y DOS representa un wildcard. Similarmente la PBX Asterisk está diseñada para conectar cualquier hardware telefónico o cualquier tipo de software de telefonía de manera transparente y consistente.

## 2.3. HIPÓTESIS Y VARIABLES

### 2.3.1. Formulación de la Hipótesis

El diseño de un sistema de adquisición y transmisión de sonidos respiratorios mediante telefonía IP, ayudará al diagnóstico de patologías respiratorias.

### 2.3.2. Operacionalización de las Variables

Tabla III: Operacionalización de las variables

Variable de estudio	Variables	Cuantitativo	Cualitativo
<b>Independiente</b>	Sistema de adquisición y transmisión de sonidos respiratorios	<ul style="list-style-type: none"> <li>• Adquisición de sonidos respiratorios</li> <li>• Transmisión de los sonidos</li> </ul>	<ul style="list-style-type: none"> <li>• Diagnosticar en tiempo real por los médicos</li> </ul>
<b>Dependiente</b>	Utilizando telefonía IP	<ul style="list-style-type: none"> <li>• Números de servidores PBX</li> <li>• Tipo de topología</li> </ul>	<ul style="list-style-type: none"> <li>• Diagnosticar en tiempo real por los médicos especialistas</li> </ul>

### 2.3.3. Contraste de Hipótesis

$H_0$  . El diseño de un sistema de adquisición y transmisión de sonidos respiratorios mediante las redes de internet usando los protocolos y códec de audio de una PBX, ayudan al diagnóstico de patologías respiratorias.

$H_1$  . El diseño de un sistema de adquisición de los sonidos respiratorios, con componentes analógicos y filtros activos para su procesamiento y transmisión mediante las redes de internet, no ayudan al diagnóstico de patologías respiratorias.



### **3.1. DESCRIPCIÓN DEL MÉTODO Y DISEÑO**

#### **3.1.1. Tipo de Investigación**

De acuerdo al problema de conocimiento a resolver y según su propósito se estableció que este proyecto es de tipo experimental.

**Área:** Ing. Electrónica

**Línea:** Telemática

**Diseño:** Aplicada

**Enfoque:** Cuantitativo

#### **3.1.2. Ámbito de Estudio**

La presente investigación será desarrollada en la ciudad de Puno.

#### **3.1.3. Nivel de Investigación**

El nivel de investigación, se realizó a nivel analítico y descriptivo, la información se analizó mediante la revisión bibliográfica. Con esta información la investigación tiene una naturaleza aplicada ya que permitirá la operacionalización de los conocimientos a fin de mostrar las prestaciones de servicios.

#### **3.1.4. Administración del Proyecto**

##### **3.1.4.1. Recursos humanos**

Ing. Ferdinand E. Pineda Ancco, asesor del proyecto y en el campo del Sistema Operativo Debian SQUEEZE v6, Dr. Tania Valencia Pacho, en la recolección de datos y una visión general en el tema de patologías respiratorias, mi hermano Hilver Charca Mamani en la cotización y compra de equipos y material de investigación, y por último al desarrollador del proyecto.

### 3.1.4.2. Presupuesto

**Tabla IV: Costo de herramientas y materiales para la propuesta**

	DESCRIPCION	PRECIO/UNIDAD	MEDIDA	TOTAL
<b>CABLE</b>	Rollo de cable UTP CAT 5E SATRA	S/. 1.50	20 METROS	S/. 30.00
<b>CONECTOR</b>	RJ 45 CAT5E	S/. 1.00	8 UNIDADES	S/. 8.00
<b>ROUTER 01</b>	TP-LINK TL-WR741ND	S/. 120.00	1 UNIDAD	S/. 120.00
<b>ROUTER 02</b>	LINKSYS WRT54GL	S/. 280.00	1 UNIDAD	S/. 280.00
<b>SWITCH</b>	D'LINK ADMINISTRABLE CAT. 5E 8 PUERTOS	S/. 70.00	1 UNIDAD	S/. 70.00
<b>02 LAPTOS</b>	Servidor Asterisk	-	2 UNIDADES	S/.3500.00
<b>TELEFONO IP</b>	Linksys Sipura modelo SPA841	S/. 200.00	2 UNIDADES	S/. 400.00
<b>CPU PENTIUM IV</b>	Servidor web y base de datos	S/. 2000.00	1 UNIDAD	S/. 1200.00
<b>COMPONENTES ELECTRONICOS</b>	Implementación del estetoscopio electrónico	-	-	S/. 200.00
<b>OTROS</b>	Gastos de envío, servicios de internet, impresión de textos	S/. 100.00	1 UNIDAD	S/. 300.00
<b>TOTAL DEL COSTO DE HERRAMIENTAS Y MATERIALES PARA LA PROPUESTA</b>				<b>S/. 6108.00</b>

### 3.1.4.3. Financiamiento

En lo que representa al financiamiento, el costo total será asumido por el ejecutor del proyecto.



**CAPITULO IV. DISEÑO, IMPLEMENTACIÓN Y ANÁLISIS DE LA  
INGENIERÍA DEL PROYECTO**

## 4.1. CIRCUITO DE ADQUISICIÓN DE SONIDOS RESPIRATORIOS

### 4.1.1. Descripción General

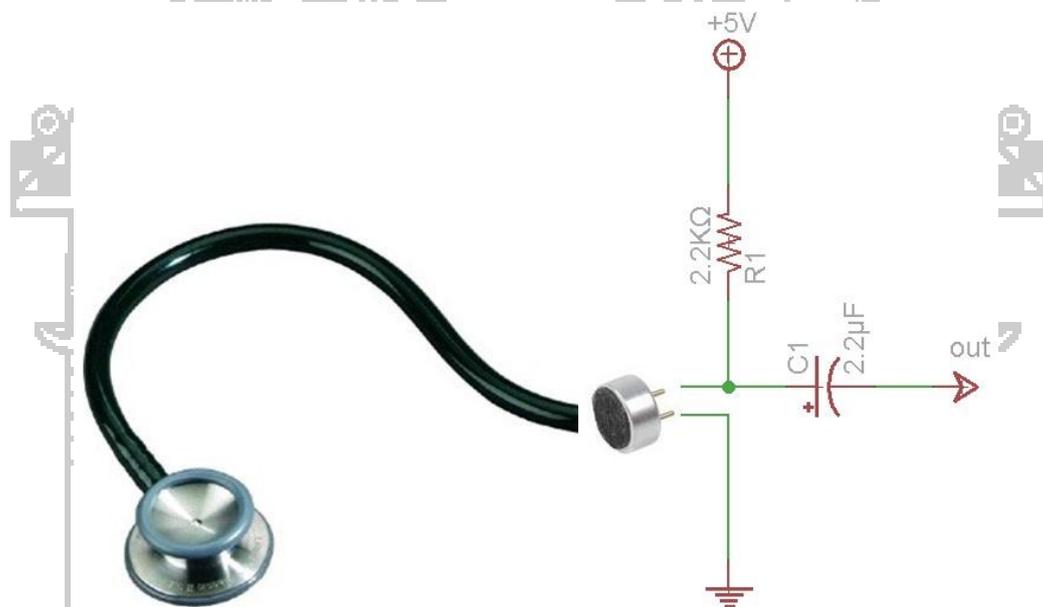
El circuito de adquisición de sonidos respiratorios implementado consiste de un sensor (micrófono), un amplificador de alta impedancia, filtros activos y ajustador de nivel compuesto por un amplificador de instrumentación; todos ellos acondicionan la señal, para después digitalizarla en la computadora. El micrófono transforma las ondas acústicas de la pared del pecho o espalda en variaciones de voltaje. El amplificador aumenta la amplitud de la señal adquirida por el micrófono. Se utiliza una combinación de filtro pasa bajas (FPB) y filtro pasa altas (FPA) en cascada, para limitar la señal a las frecuencias de interés. Dado que el ancho de banda comúnmente usado para sonidos respiratorios va de 60Hz a 2kHz, para auscultaciones en el pecho (sonidos pulmonares), y de 60Hz a 4kHz, para cuando son adquiridos en la tráquea, el propósito de utilizar un FPA es reducir los ruidos producidos por el corazón, los músculos y otras fuentes de baja frecuencia (generados por contacto o el ambiente). El FPB es de antienmascaramiento y se utiliza para evitar el traslape espectral. El ajustador de nivel comunica las señales adquiridas a la entrada de audio de PC para luego ser procesados y enviados a la red, también tiene salida para los audífonos, esto para el auscultador. En la Figura 24, se muestra el diagrama de bloques del sistema mencionado.



**Figura 24. Diagrama a bloques del circuito de adquisición de sonidos respiratorios**

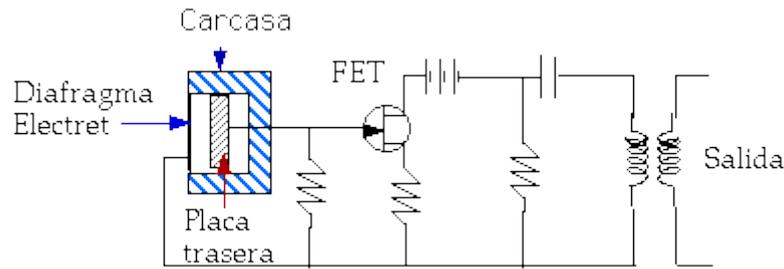
#### 4.1.2. Sensor

Para lograr oír los sonidos respiratorios, se va a utilizar la campana de un estetoscopio unido a un tubo de PVC. Este tubo irá a su vez unido a un micrófono. En nuestro caso utilizaremos un micrófono “electret”, Un Micrófono Electret tiene como característica su capacidad de mantener carga sin necesidad de una fuente de polarización, por lo cual tiene cada vez mayor popularidad por razones de economía y su alta impedancia de entrada. La alta impedancia en la entrada implica que las corrientes de entrada en el circuito son menores que en baja impedancia. Esto implica que la relación señal ruido del circuito es mejor.



**Figura 25. Primera parte del estetoscopio digital**

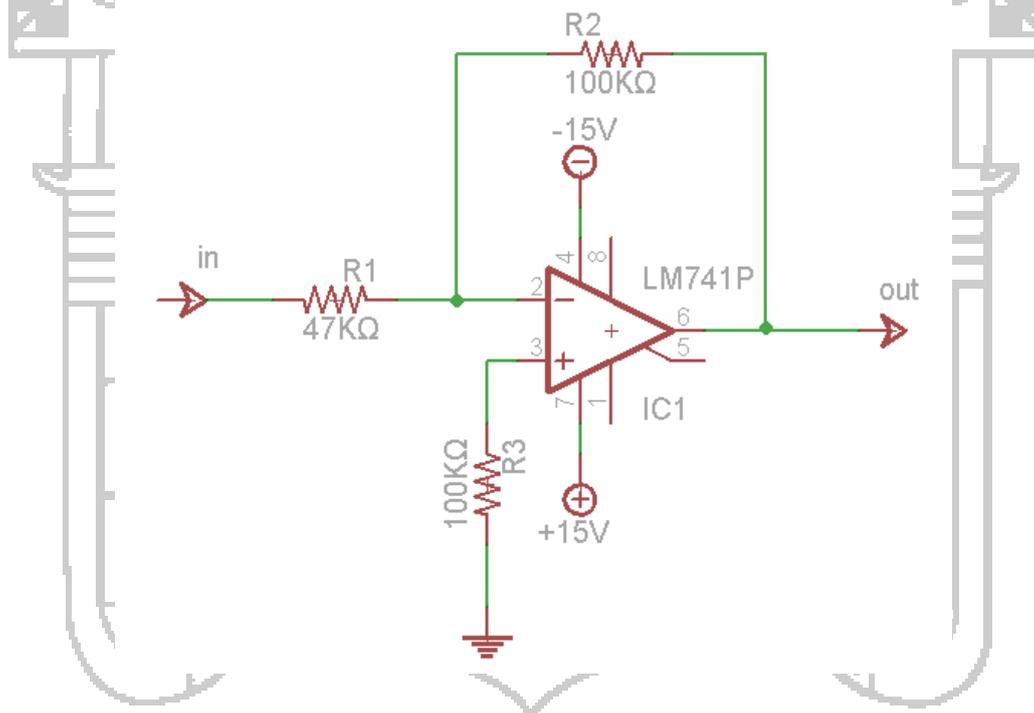
Se recoge el sonido a través de la membrana de la campana y que este sonido pase a través del tubo de PVC hasta llegar al micrófono. Aquí comienza la parte electrónica del estetoscopio. Con el micrófono se convierte la señal de sonido a una señal eléctrica, para ello tenemos que polarizar el micrófono.



**Figura 26. Esquemático del micrófono**

#### 4.1.3. Amplificador de Alta Impedancia

El circuito consta de un amplificador de alta impedancia. El montaje correspondiente está construido a partir de un amplificador operacional UA741, configurado como inversor que se muestra en la Figura 27. Este amplificador tiene una ganancia que depende de las resistencias conectadas.



**Figura 27. Montaje amplificador inversor**

Para la caracterización del amplificador inversor en el montaje que nos ocupa, las ecuaciones del dispositivo serían:

$$V^+ = V^- = 0 \tag{1}$$

$$I^+ = I^- = 0 \quad (2)$$

Teniendo en cuenta esto, obtenemos las ecuaciones del amplificador inversor.

$$I_{R1} = \frac{V_{in} - V^-}{R1} \quad (3)$$

$$I_{R2} = \frac{V^- - V_{out}}{R2} \quad (4)$$

Despejando 1 en 3 y 4 obtenemos:

$$I_{R1} = \frac{V_{in}}{R1} \quad (5)$$

$$I_{R2} = \frac{-V_{out}}{R2} \quad (6)$$

Sabiendo que  $I_{R1} = I_{R2}$  y reemplazando las ecuaciones 5 y 6 obtenemos:

$$\frac{V_{in}}{R1} = \frac{-V_{out}}{R2} \quad (7)$$

Entonces:

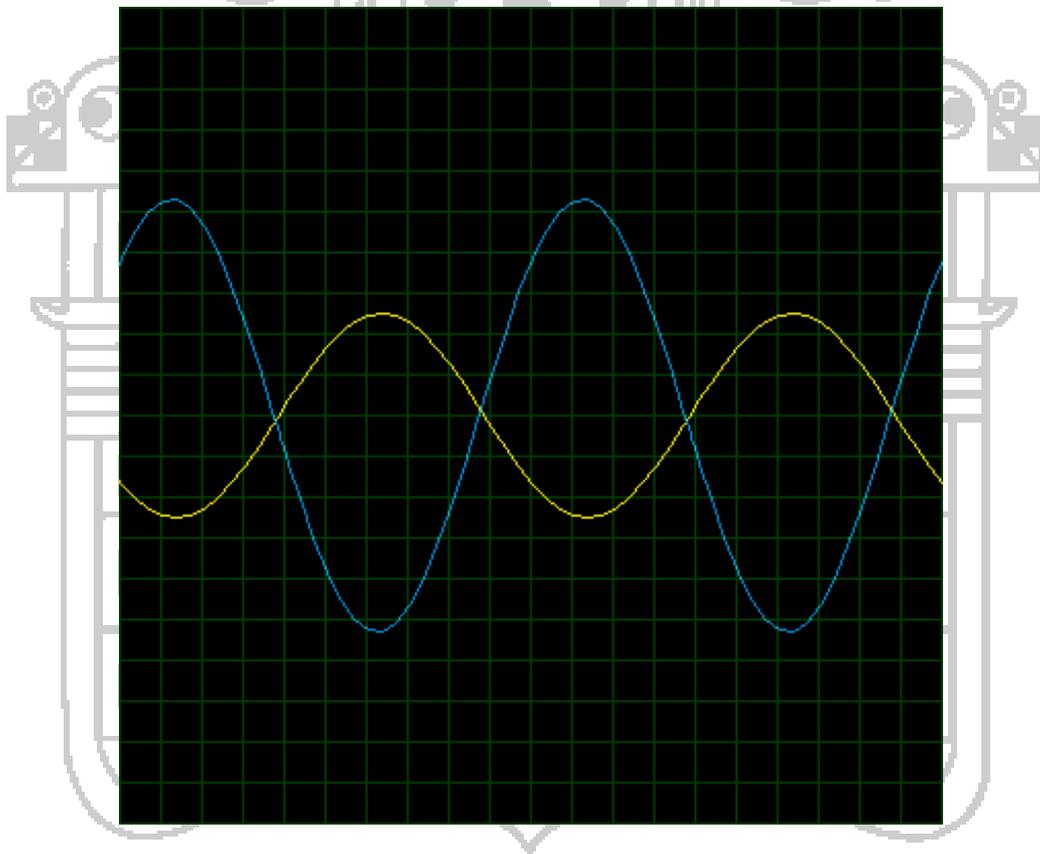
$$\frac{V_{out}}{V_{in}} = -\frac{R2}{R1} \quad (8)$$

Luego asignamos a R2 el valor de 100kΩ y R1 el valor de 47kΩ para obtener una ganancia cercana a 2.

$$A_V = \frac{V_{out}}{V_{in}} = -\frac{R_2}{R_1} = -\frac{100K\Omega}{47K\Omega}$$

$$A_V = -2.127 \quad (9)$$

A continuación procedemos a la simulación del amplificador inversor con la ayuda de Proteus 7.9 SP1. En la Ilustración 28, se aprecian dos señales, una señal amarilla en el canal 1 del osciloscopio que es la señal de entrada y una señal de color azul claro en el canal 2 que es la señal de salida. Se pueden ver la tensión pico-pico de cada señal y su frecuencia. También se puede ver el desfase de  $180^\circ$  provocado por el amplificador inversor.



**Figura 28. Simulación Experimental del Amplificador 10 Hz**

#### **4.1.4. Filtros Activos**

Para el filtrado de la señal, se va a utilizar filtros activos. Se ha hecho esta elección para este proyecto en concreto debido a que las ventajas que tienen frente a los pasivos superan los inconvenientes. Como ventajas tenemos que:

- Permiten eliminar las inductancias que, en bajas frecuencias, son voluminosas pesadas y caras.
- Se pueden diseñar filtros muy complejos mediante el diseño de etapas simples.
- Proporcionan una gran ganancia a la entrada de la señal, lo que en nuestro proyecto es muy importante ya que estamos trabajando con niveles de señales muy bajos.

Por el contrario, los inconvenientes serían:

- La exigencia de una fuente de alimentación.
- Su respuesta en frecuencia viene limitada por la respuesta del amplificador operacional.
- Es imposible su aplicación en sistemas de alta potencia.

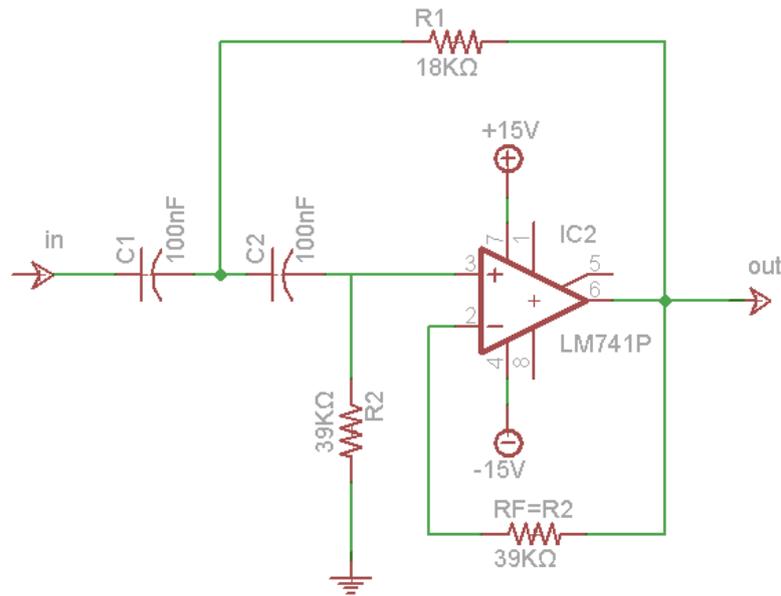
Por otra parte, no vamos a tener ningún problema de sistemas de alta potencia. Finalmente y sabiendo la finalidad del proyecto, se utilizan unos amplificadores operacionales con una respuesta lo suficientemente buena como para que tengamos la señal deseada.

Por ello, se van a utilizar filtros activos. El uso de los amplificadores operacionales permite tener una alta resistencia de entrada y una baja resistencia de salida entre otras características.

#### 4.1.4.1. Filtro Paso Alto

Para el propósito de este proyecto el filtro pasa alto necesita una frecuencia de corte  $f_n = 60\text{Hz}$ , como ya lo vimos en el apartado 2.1.3. Para atenuar todo lo posible las frecuencias inferiores y permitir el paso de frecuencias altas.

Las características de este filtro van a depender de no superar un filtro de orden 2, ya que esto puede suponer el uso de amplificadores adicionales, añadiendo ruido, como se muestra en la figura 29.



**Figura 29. Diseño del Filtro Paso Alto**

Los filtros de tipo sellen-Key son sistemas de segundo orden con dos polos en su función de transferencia. El circuito consta de un amplificador no inversor, dos resistencias y dos condensadores.

Las expresiones que consideramos para el caso particular de filtro pasa alto, tenemos:

$$C1 = C2 = C \tag{10}$$

$$R2 \neq R1 \tag{11}$$

$$f_n = \frac{1}{2\pi(R1)C\sqrt{2}} \tag{12}$$

$$Q = \frac{1}{2}\sqrt{R2/R1} \tag{13}$$

Para la configuración en Butteworth partimos que, el factor de selectividad  $Q = 0.707$  en la ecuación 13,  $R2$  con respecto a  $R1$  tenemos:

$$R2 = 2(R1) \quad (14)$$

Si consideramos  $C = 100nF$ , y modificando la ecuación 12, calculamos el valor de R1:

$$R1 = \frac{1}{2\pi f_n C \sqrt{2}} \quad (15)$$

$$R1 = \frac{1}{2 \cdot \pi \cdot 60 \cdot 100 \cdot 10^{-9} \cdot \sqrt{2}} \quad (16)$$

$$R1 = 18.77K\Omega \cong 18K\Omega \quad (17)$$

Remplazando en la ecuación 14, obtenemos el valor de R2:

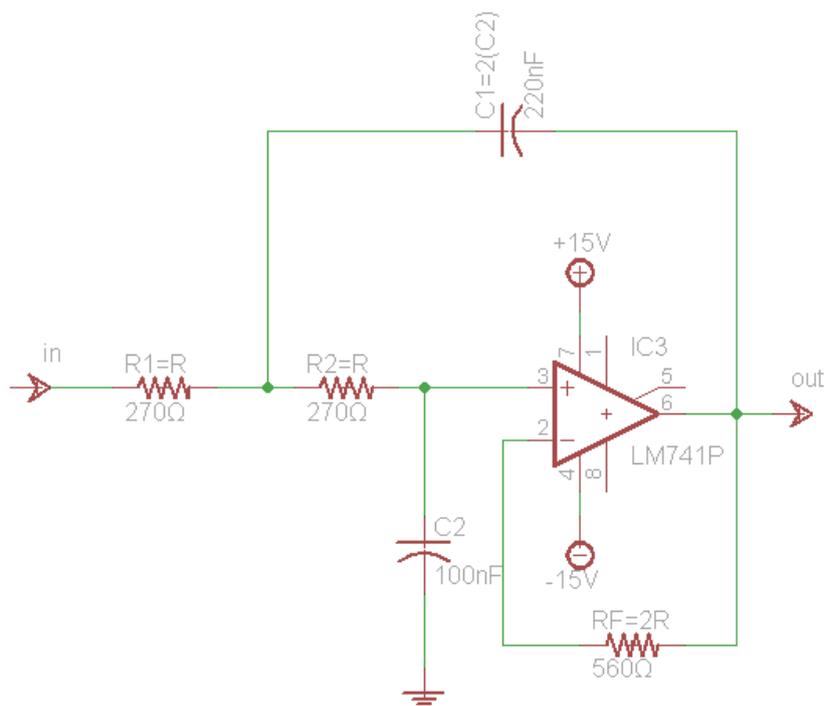
$$R2 = 37,52K\Omega \cong 39K\Omega \quad (18)$$

Se elige el valor de  $R2 = 39K\Omega$  por ser este comercial en una tienda de componentes electrónicos.

#### 4.1.4.2. Filtro Paso Bajo

Este filtro paso bajo, al igual que en el caso anterior, va a tener una frecuencia de corte de  $f_n = 4KHz$ , un filtro Butteworth, debido a que no nos interesa introducir ganancia en el filtro y este filtro posee esta característica. También es conocido como filtro máximamente plano.

Además de contar con el esquema de Butteworth de orden 2 y la configuración filtro pasa bajo de Sallen-Key, como se muestra en la ilustración 30.



**Figura 30. Diseño del Filtro Paso Bajo**

Para el diseño particular de filtro pasa bajo consideraremos las siguientes fórmulas:

$$C1 \neq C2 \tag{19}$$

$$R1 = R2 = R \tag{20}$$

$$f_n = \frac{1}{2\pi R(C2)\sqrt{2}} \tag{21}$$

$$Q = \frac{1}{2}\sqrt{C1/C2} \tag{22}$$

$$RF = 2R \tag{23}$$

Al igual que antes, el filtro Butterworth tiene  $Q = 0.707$ , considerando éste parámetro  $C1 = 2(C2)$ .

Si consideramos:

$$C2 = 100nF \quad (24)$$

Entonces:

$$C1 = 200nF \cong 220nF \quad (25)$$

Modificando la ecuación 21, calculamos el valor de R:

$$R = \frac{1}{2\pi f_n (C2)\sqrt{2}} \quad (26)$$

$$R = \frac{1}{2 \cdot \pi \cdot 4 \cdot 10^3 \cdot 100 \cdot 10^{-9} \cdot \sqrt{2}} \quad (27)$$

$$R = 281,35\Omega \cong 270\Omega \quad (28)$$

Reemplazando  $R = 281,35\Omega$  en la ecuación 23 obtenemos el valor de  $RF$ :

$$RF = 2R$$

$$RF = 2(281,35\Omega) \quad (29)$$

$$RF = 562,697\Omega \cong 560\Omega \quad (30)$$

#### 4.1.5. Ajustador de Nivel

El montaje del ajustador de nivel consta de un circuito integrado INA129P de la Burr-Brown, que es un Amplificador de Instrumentación de alta precisión (el fabricante lo recomienda para aplicaciones biomédicas). Este integrado es un

Amplificador de Instrumentación de propósito general, que ofrece una exactitud excelente.

La versatilidad de su diseño y su pequeño tamaño lo hacen ideal para una amplia gama de aplicaciones. Una sola resistencia externa hace variar la ganancia de 1 a 10000. Posee una protección interna a la entrada que resiste hasta  $\pm 40V$  sin dañarse.

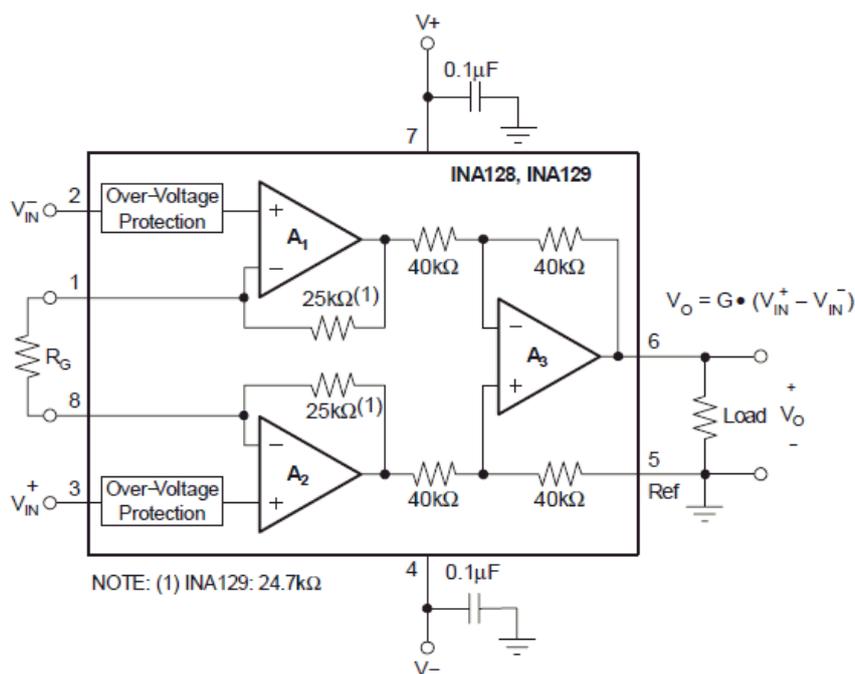
El INA129P es fabricado con láser lo cual logra un muy bajo voltaje de Offset ( $50\mu V$ ), con una variación ( $0.5\mu V/^{\circ}C$ ) ofreciendo alto rechazo en modo común ( $120dB$  a  $G \geq 1000$ ).

Operan con alimentaciones tan bajas como  $\pm 2.25V$  a  $\pm 18V$ , permitiendo uso de baterías y una corriente inmóvil de  $3mA$  máximo. El INA129P está disponible en encapsulado de 8 pines de plástico con un rango de temperatura  $-40^{\circ}C$  a  $+85^{\circ}C$ .

#### 4.1.5.1. Diagrama Interno del INA129.

La figura 31 representa el diseño clásico utilizado por la mayoría de los amplificadores de instrumentación. La salida del amplificador operacional es un amplificador diferencial con ganancia de tensión unidad. Las resistencias utilizadas en esta etapa de salida son normalmente menos de un  $\pm 0.001\%$  de diferencia entre ellas gracias a los cortes realizados con láser. La figura 31 muestra las conexiones básicas requeridas para el funcionamiento del INA129. Aplicaciones con suministros de voltaje de alta impedancia o ruidosos pueden requerir condensadores de desacoplo cerca del dispositivo.

La salida (pin 6) es referenciada a tierra, normalmente el pin 5 se conecta a tierra. La ganancia del INA129P, viene dada por la relación que el fabricante especifica en el datasheet:



**Figura 31. Diagrama Interno INA129P**

Tomando en cuenta los valores sugeridos por el fabricante (Burr-Brown). Se selecciona el amplificador de instrumentación de precisión y de baja potencia INA129P.

$$\text{INA128: } G = 1 + \frac{50k\Omega}{R_G} \quad \text{INA129: } G = 1 + \frac{49.4k\Omega}{R_G}$$

DESIRED GAIN (V/V)	INA128		INA129	
	R <sub>G</sub> (Ω)	NEAREST 1% R <sub>G</sub> (Ω)	R <sub>G</sub> (Ω)	NEAREST 1% R <sub>G</sub> (Ω)
1	NC	NC	NC	NC
2	50.00k	49.9k	49.4k	49.9k
5	12.50k	12.4k	12.35k	12.4k
10	5.556k	5.62k	5489	5.49k
20	2.632k	2.61k	2600	2.61k
50	1.02k	1.02k	1008	1k
100	505.1	511	499	499
200	251.3	249	248	249
500	100.2	100	99	100
1000	50.05	49.9	49.5	49.9
2000	25.01	24.9	24.7	24.9
5000	10.00	10	9.88	9.76
10000	5.001	4.99	4.94	4.87

NC: No Connection

**Figura 32. Tabla de datos del INA129P**

Donde G es la ganancia deseada.

R<sub>G</sub> es una resistencia que determina la ganancia.

La resistencia que determina la ganancia se calcula de acuerdo a la ecuación R<sub>G</sub>. Para calcular las ganancias entre 100 y 200 tenemos:

$$R_G = \frac{49.4K\Omega}{G - 1} \tag{31}$$

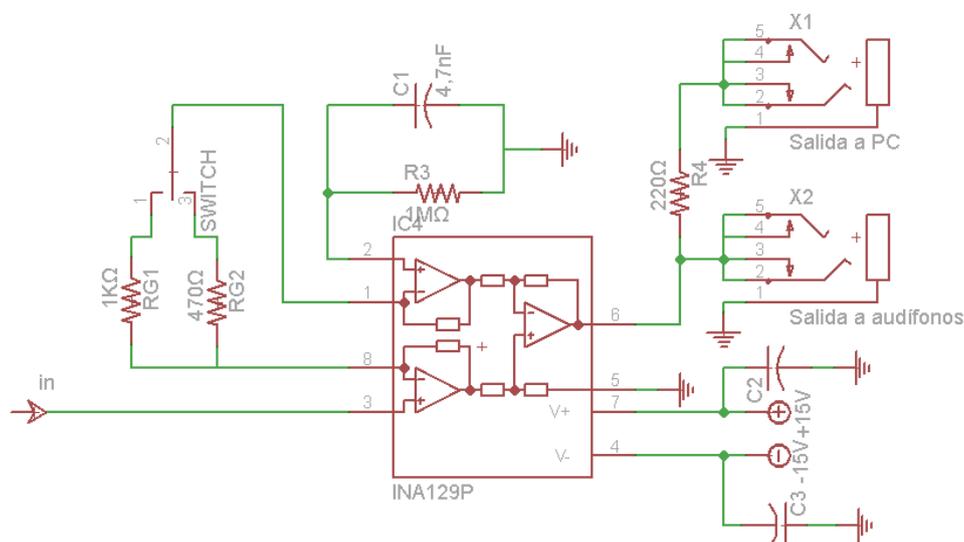
Si G=100, entonces:

$$R_G = 498.98\Omega \tag{32}$$

Si G=50, entonces:

$$R_G = 1008.16\Omega \cong 1K\Omega \tag{33}$$

Es aconsejable usar resistencias de precisión y para mayor comodidad en los cálculos ver la tabla de datos especificado en el datasheet.

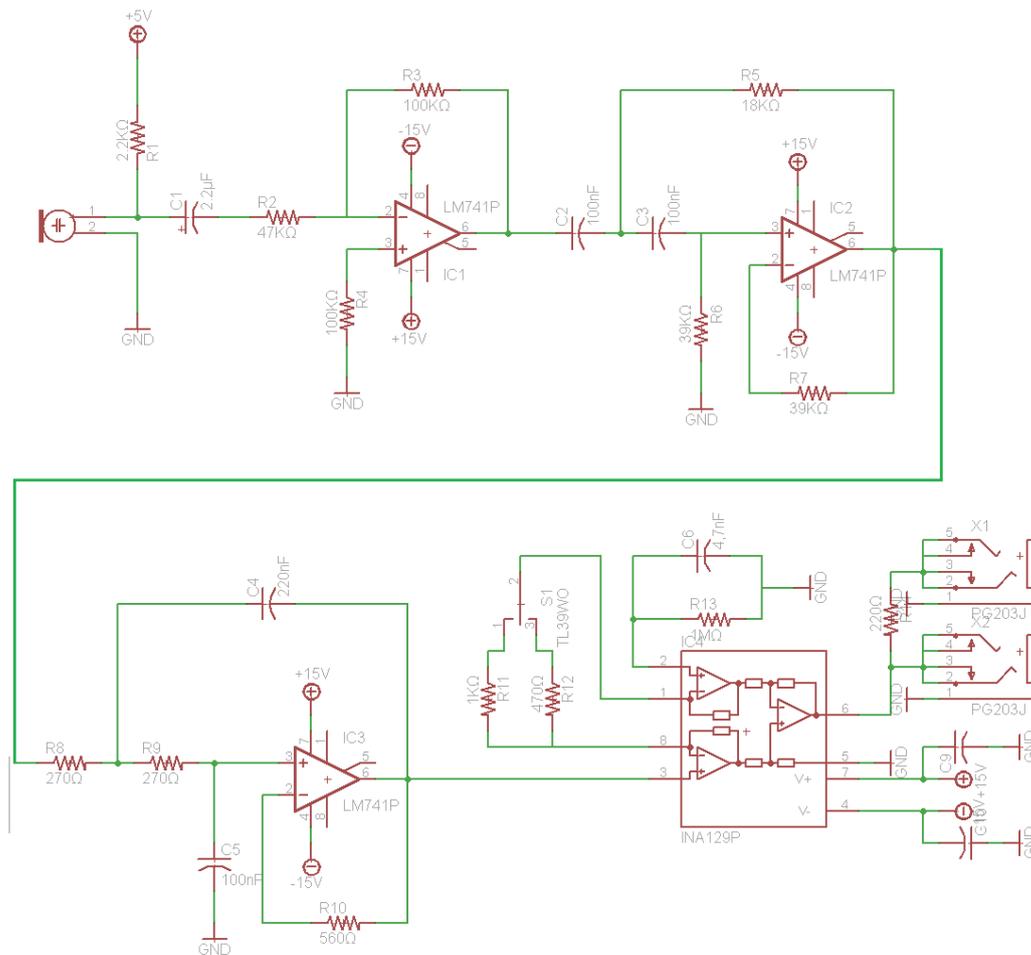


**Figura 33. Diseño del Ajustador de Nivel**

#### 4.1.6. Diseño Completo

Finalmente, vamos a ver el circuito completo. Para ello emplearemos la herramienta que hemos estado utilizando durante este proyecto fin de carrera, Start Cadsoft Eagle software.

Como se ve en la figura 34, se ha conectado el amplificador inversor, como la primera etapa, seguido de los dos filtros. El filtro paso alto y el paso bajo, los cuales funcionan como un paso banda. El motivo de esta colocación, como ya se ha explicado anteriormente es que al introducir el amplificador de alta impedancia como primera etapa del circuito mejora la relación señal ruido del sistema. Como última etapa el ajustador de nivel, que está compuesto por un Amplificador de Instrumentación de alta precisión con el fin de obtener la ganancia deseada.



**Figura 34. Diseño Completo**



**Figura 35. Ubicación de los componentes en 3D**

## 4.2. Medio de Transmisión

La plataforma de soporte físico va depender de las redes interconectadas, ya sea una red LAN o una red WAN como se muestra en la figura 36. Para el desarrollo de este proyecto, se implementó a modo de prototipo una red LAN, compuesta por tres ordenadores. Cada ordenador tiene instalado los sistemas operativos Linux Debian, así como Windows XP. Siendo usado para el proyecto exclusivamente Linux durante todos los casos de uso.

Para la realización de los casos prácticos de los que está compuesto el proyecto se hará uso de la centralita Asterisk, cuya instalación y configuración se detalla a continuación:

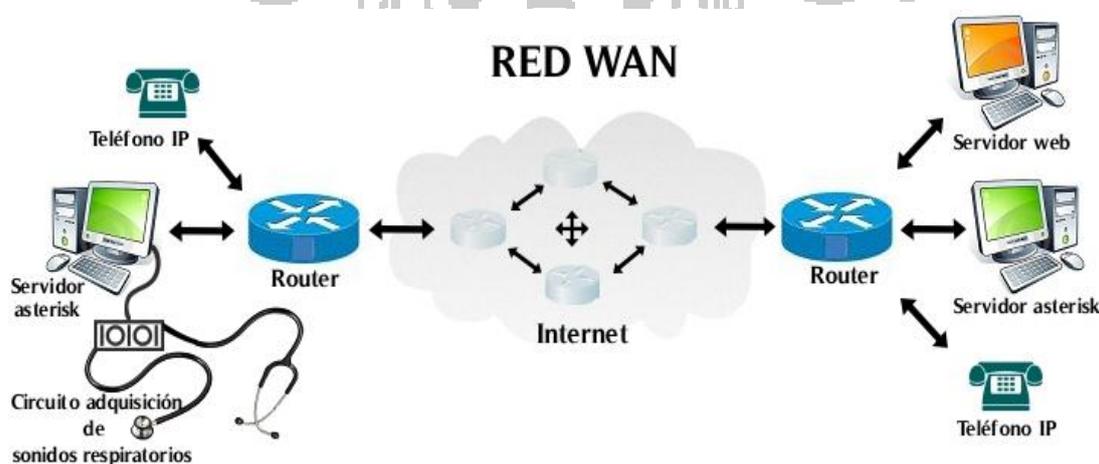
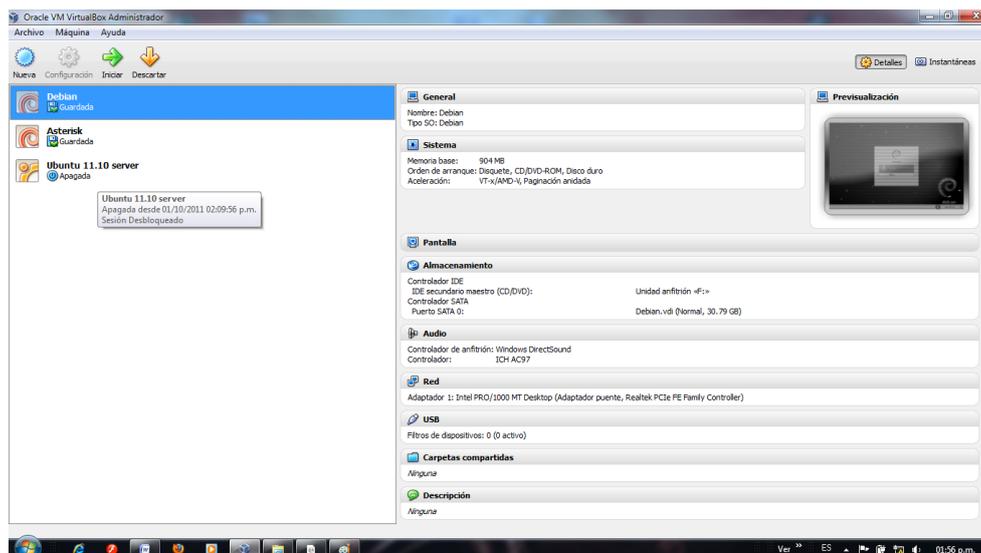


Figura 36. Diseño de una Red WAN

### 4.2.1. Instalación de Linux Debian Wheesy

Usamos Debian con Kernel 2.6 para la instalación de Asterisk con Linux. Escogemos esta distribución por la gran aceptación que tiene como una de sus ventajas el apt-get y por ser una de las distribuciones soportadas por la Digium. Abajo siguen las instrucciones de instalación de Asterisk a partir de cero, en mi caso use una máquina virtual (VirtualBOX) para un primer servidor de asterisk, y una maquina física para un segundo servidor asterisk, la instalación es igual o semejante en ambos casos, dependiendo de la versión que se está implementando.



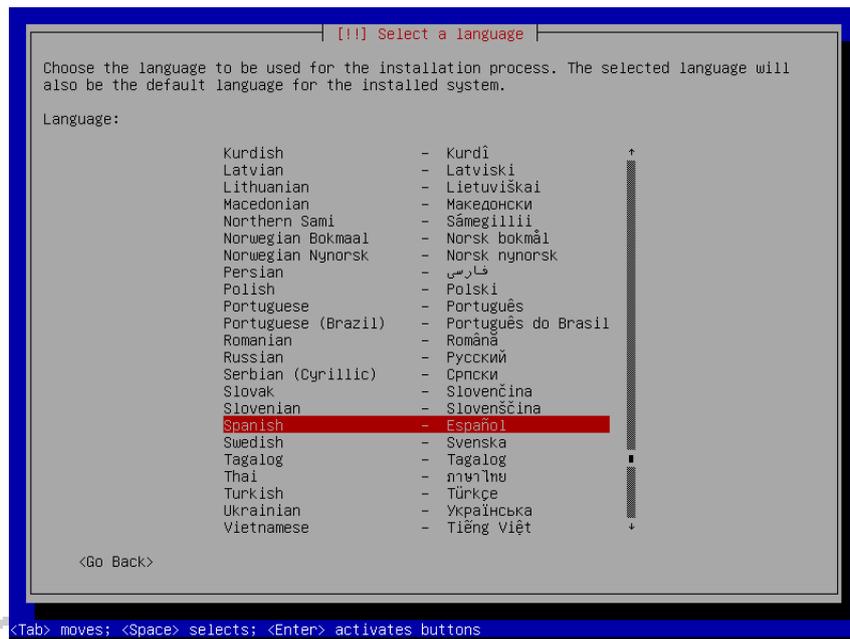
**Figura 37. VirtualBOX**

**Paso 01:** Coloque el CD de Debian en la lectora de CDROM e inicialice el boteo de su PC desde allí con opción install empieza tu mundo en Linux.



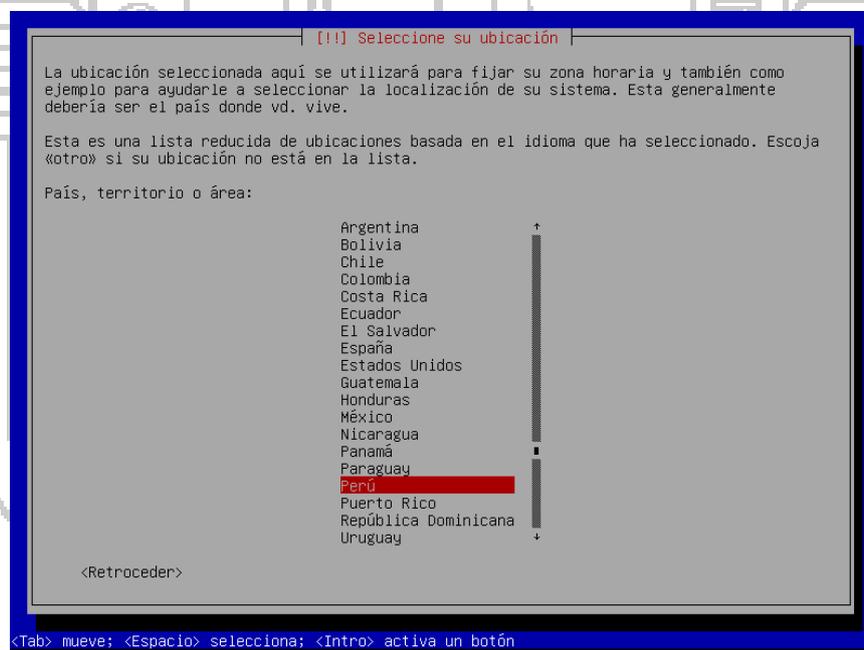
**Figura 38. Ejecución de Debian linux**

**Paso 02:** Seleccione el idioma para la instalación.



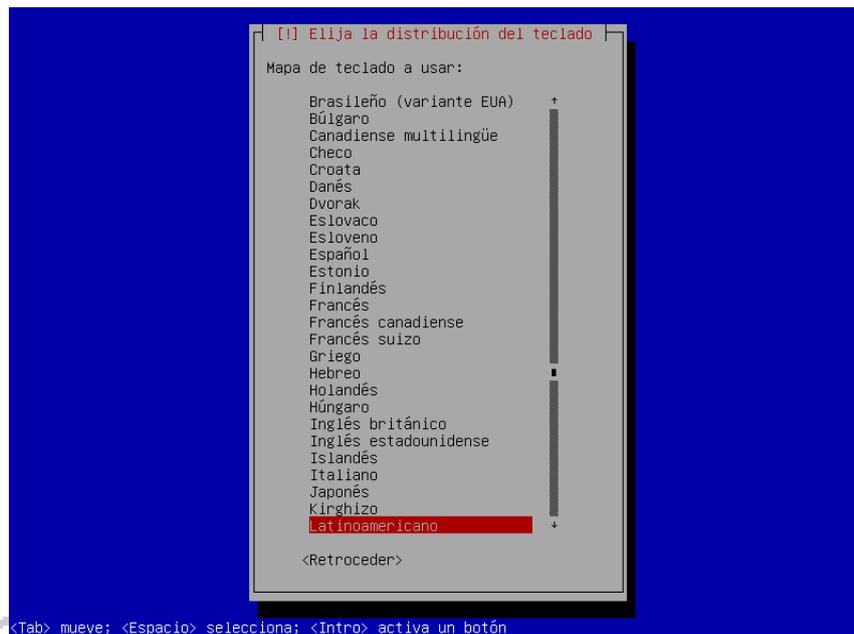
**Figura 39.** Selección del lenguaje

**Paso 03:** Seleccione el país de idioma seleccionado anteriormente.



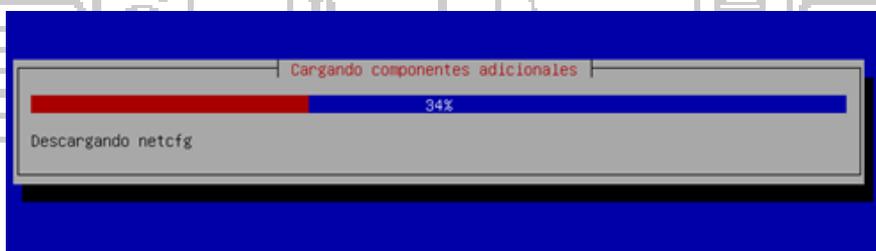
**Figura 40.** Selección de ubicación

**Paso 04:** Escoja ahora el mapa de teclado correspondiente.



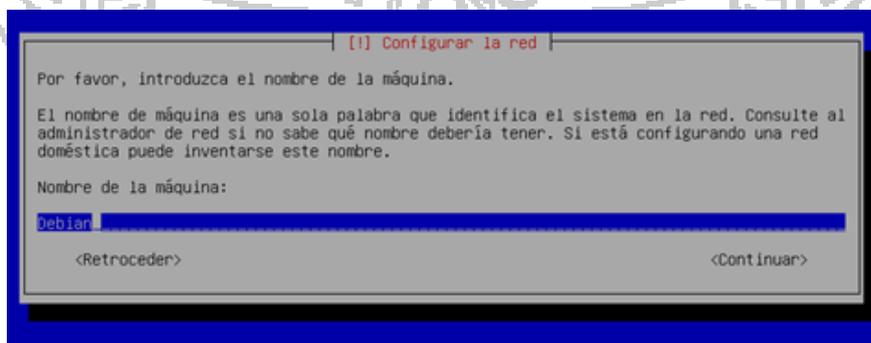
**Figura 41. Selección de teclado**

Esperamos que cargue.



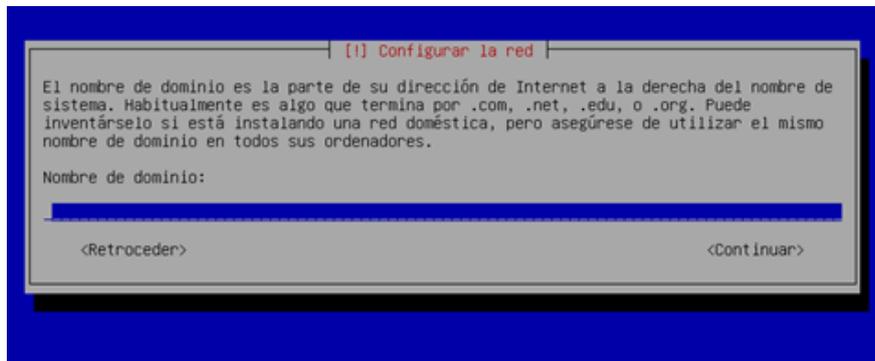
**Figura 42. Reconocimiento del Hardware**

**Paso 05:** Escriba el nombre de host de la máquina.



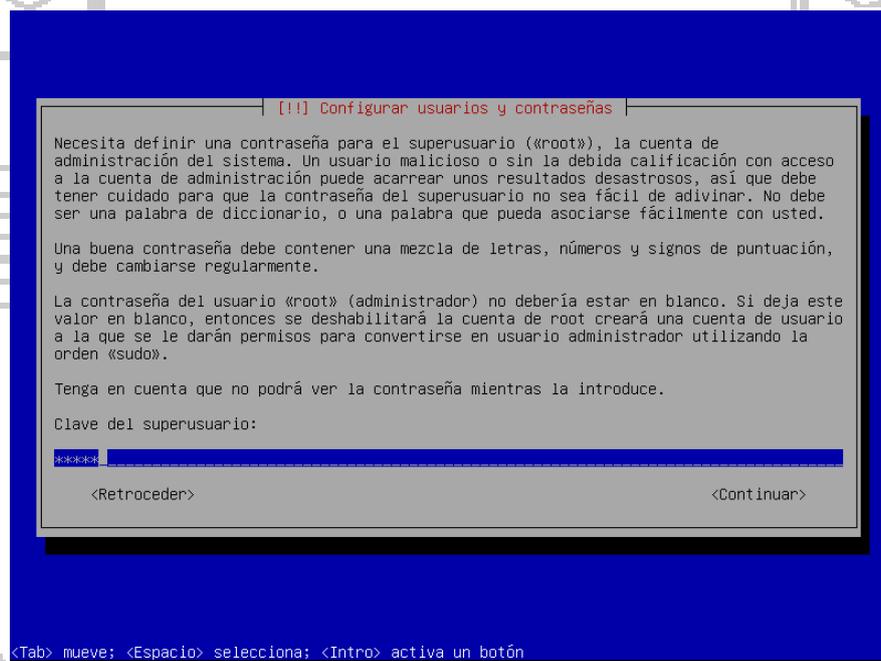
**Figura 43. Nombre de la maquina**

**Paso 06:** Escriba el dominio al cual este equipo es parte, en mi caso lo dejo en blanco.



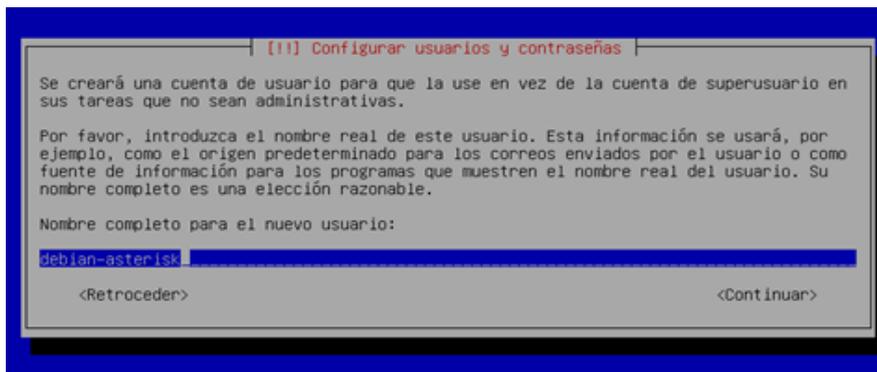
**Figura 44. Nombre de dominio**

**Paso 07:** Escriba su contraseña y confirme, que más adelante lo usaremos en el terminal como superusuario.



**Figura 45. Clave de usuario**

**Paso 08:** Escriba el nombre completo en nuevo usuario y contraseña.



[!!] Configurar usuarios y contraseñas

Se creará una cuenta de usuario para que la use en vez de la cuenta de superusuario en sus tareas que no sean administrativas.

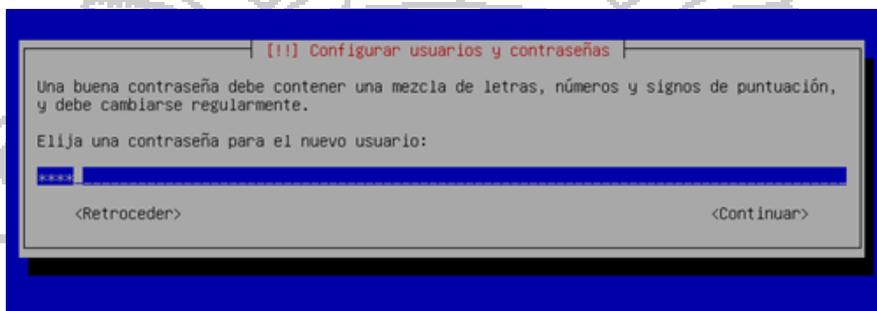
Por favor, introduzca el nombre real de este usuario. Esta información se usará, por ejemplo, como el origen predeterminado para los correos enviados por el usuario o como fuente de información para los programas que muestren el nombre real del usuario. Su nombre completo es una elección razonable.

Nombre completo para el nuevo usuario:

debian-asterisk

<Retroceder> <Continuar>

**Figura 46. Nombre de usuario**



[!!] Configurar usuarios y contraseñas

Una buena contraseña debe contener una mezcla de letras, números y signos de puntuación, y debe cambiarse regularmente.

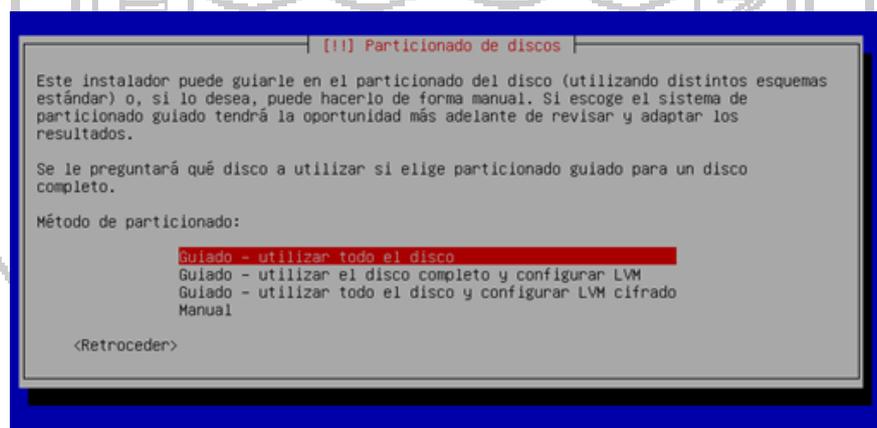
Elija una contraseña para el nuevo usuario:

\*\*\*\*

<Retroceder> <Continuar>

**Figura 47. Contraseña de usuario**

**Paso 09:** Será hecha la configuración de particionado del disco del equipo, apenas confirme las opciones presentadas.



[!!] Particionado de discos

Este instalador puede guiarle en el particionado del disco (utilizando distintos esquemas estándar) o, si lo desea, puede hacerlo de forma manual. Si escoge el sistema de particionado guiado tendrá la oportunidad más adelante de revisar y adaptar los resultados.

Se le preguntará qué disco a utilizar si elige particionado guiado para un disco completo.

Método de particionado:

- Guiado - utilizar todo el disco
- Guiado - utilizar el disco completo y configurar LVM
- Guiado - utilizar todo el disco y configurar LVM cifrado
- Manual

<Retroceder>

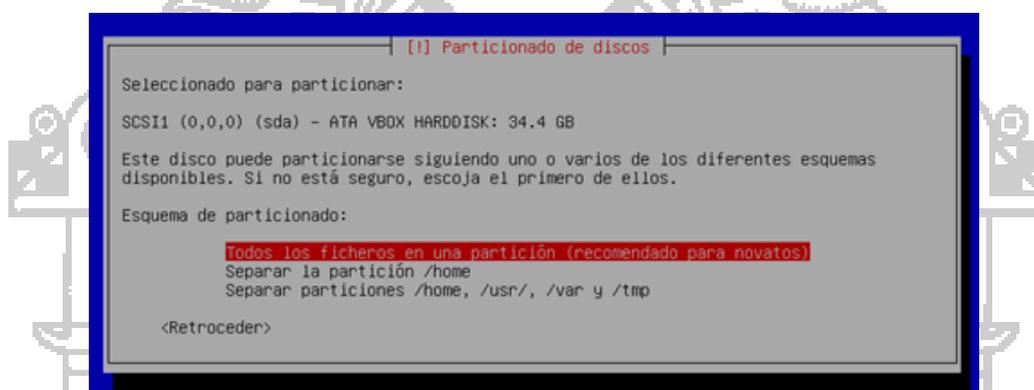
**Figura 48. Particionado de disco guiado**

**Paso 10:** Confirme que usted va a formatear todo el disco



**Figura 49. Particionado de disco**

**Paso 11:** Confirme que todos los archivos estarán en una partición.

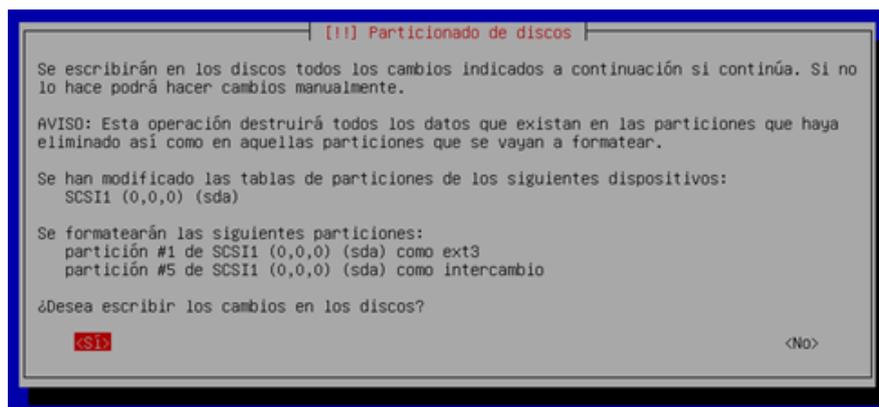


**Figura 50. Particionado de disco guiado recomendado para novatos**

**Paso 12:** finalice y acepte las opciones de particionado.

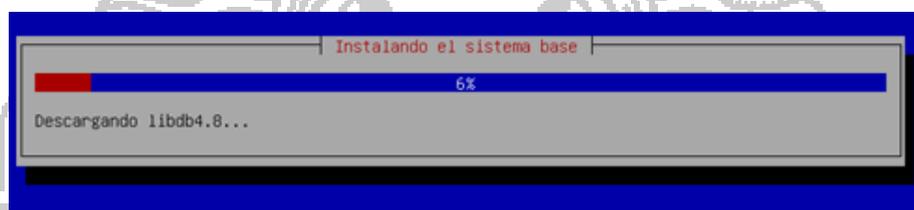


**Figura 51. Finalizar particionado de disco guiado**



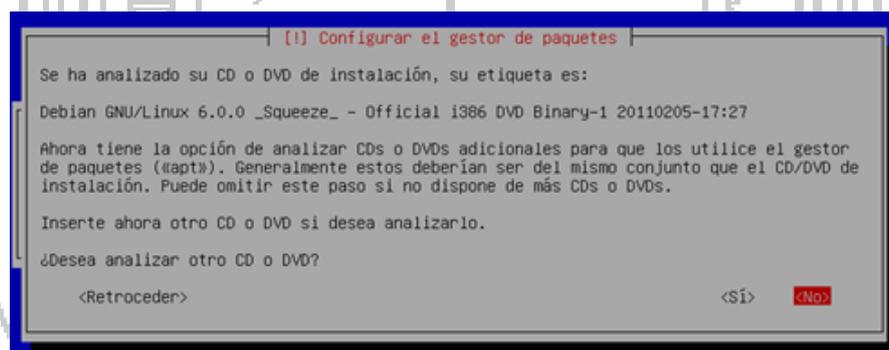
**Figura 52. Confirmar particionado de disco guiado**

**Paso 13:** Una vez confirmado espere.



**Figura 53. Instalando el sistema base**

**Paso 14:** En gestor de paquetes pide analizar otro DVD, aquí <enter> en la opción no.

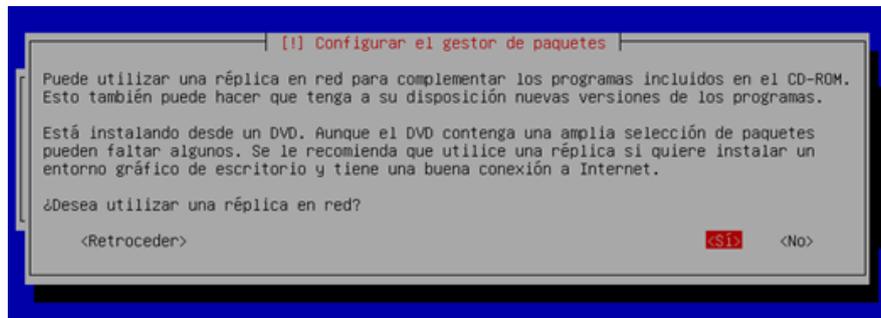


**Figura 54. Configurando gestor de paquetes**

**Paso 15:** Seleccione dependiendo si desea instalar desde los servidores de Debian usando internet, la opción SI o desde el mismo DVD la opción NO.

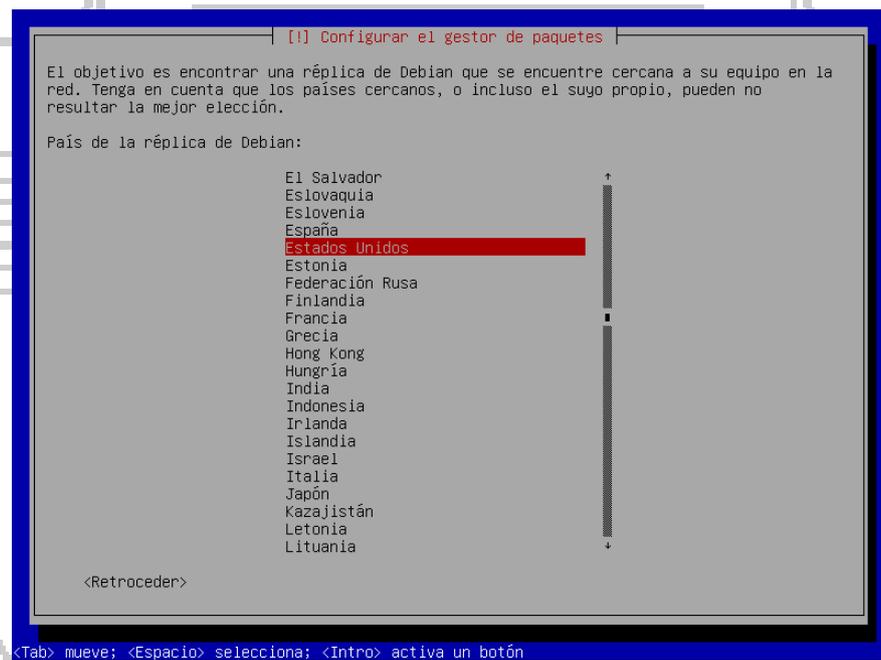
Tener en cuenta de que versión de instalador que se esté usando, puede ser versión Netis el más liviano que para su instalación es necesario la conexión a

internet, como también los de colección en DVD que son 8, pero es el primero y necesario si deseamos instalar Debian.



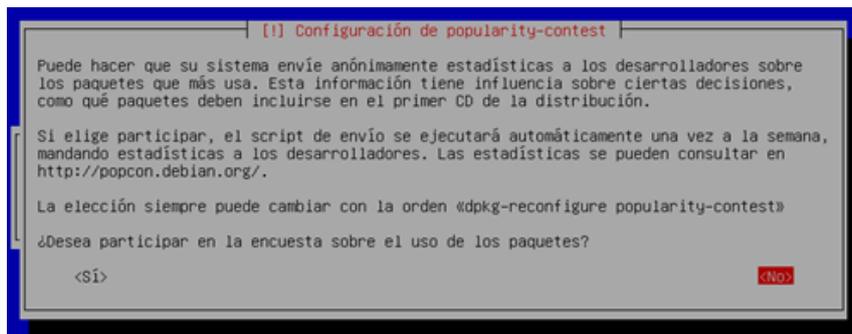
**Figura 55. Configurando réplica de red**

**Paso 16:** Si en el paso anterior presionaste la opción SI, entonces enlázale a un servidor de Estados Unidos, por experiencia, o como también el que tu desees.



**Figura 56. Configurando réplica de red por el usuario**

**Paso 17:** En popularidad presione NO.



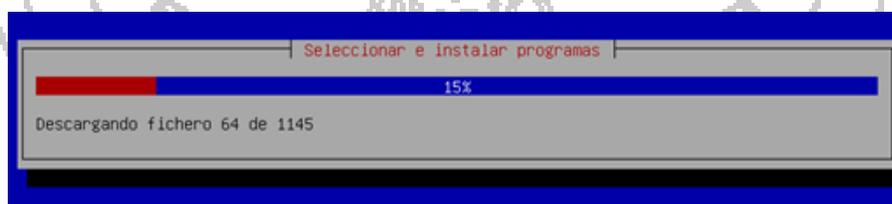
**Figura 57.** Configuración de popularidad

**Paso 18:** Ya viene seleccionado el entorno gráfico y utilidad estándar del sistema, si deseas agregar algo más, puedes hacerlo.



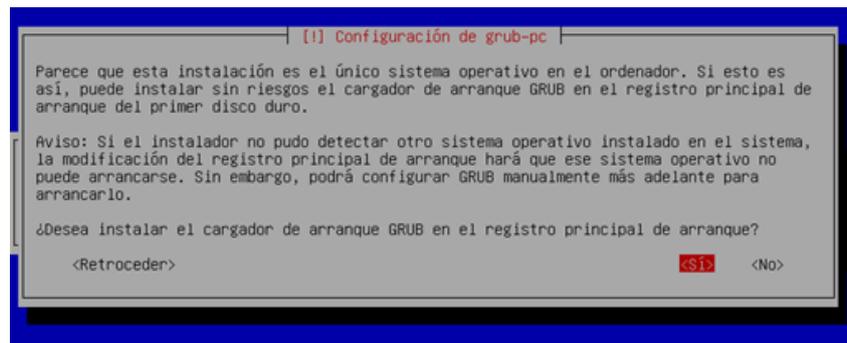
**Figura 58.** Selección de programas a instalar

**Paso 19:** Espera Tomando una, dos, tres y hasta cuatro tazas de café.



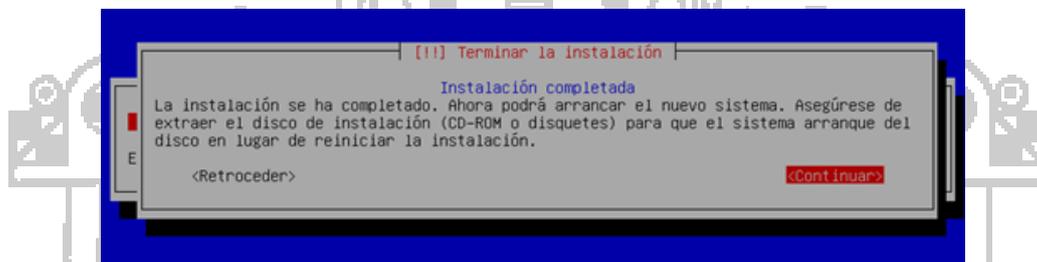
**Figura 59.** Instalando programas del sistema operativo

**Paso 20:** Acepte la instalación de GRUB.



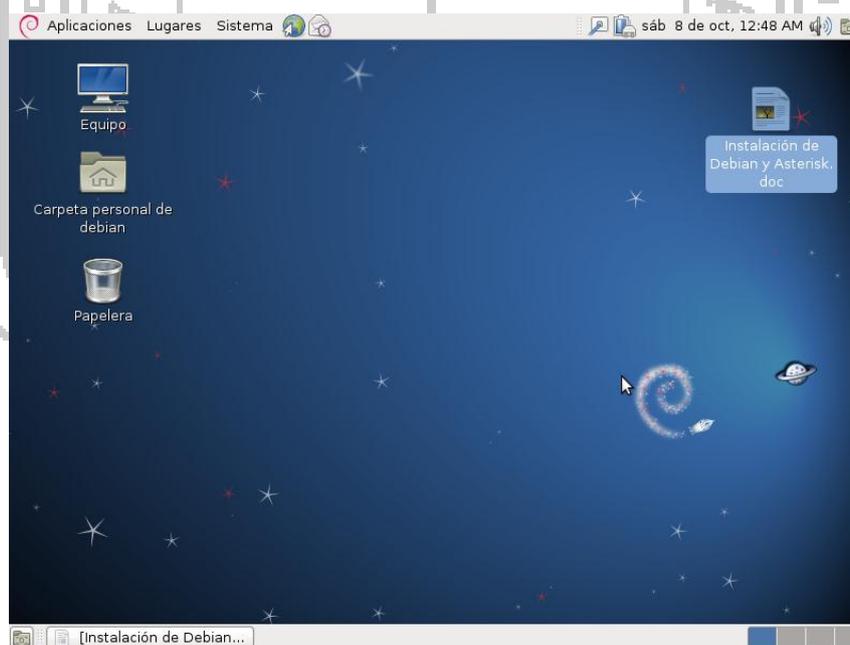
**Figura 60. Configuración de GRUB**

**Paso 21:** La instalación está completa, digite <enter> para continuar.



**Figura 61. Finalizando la instalación**

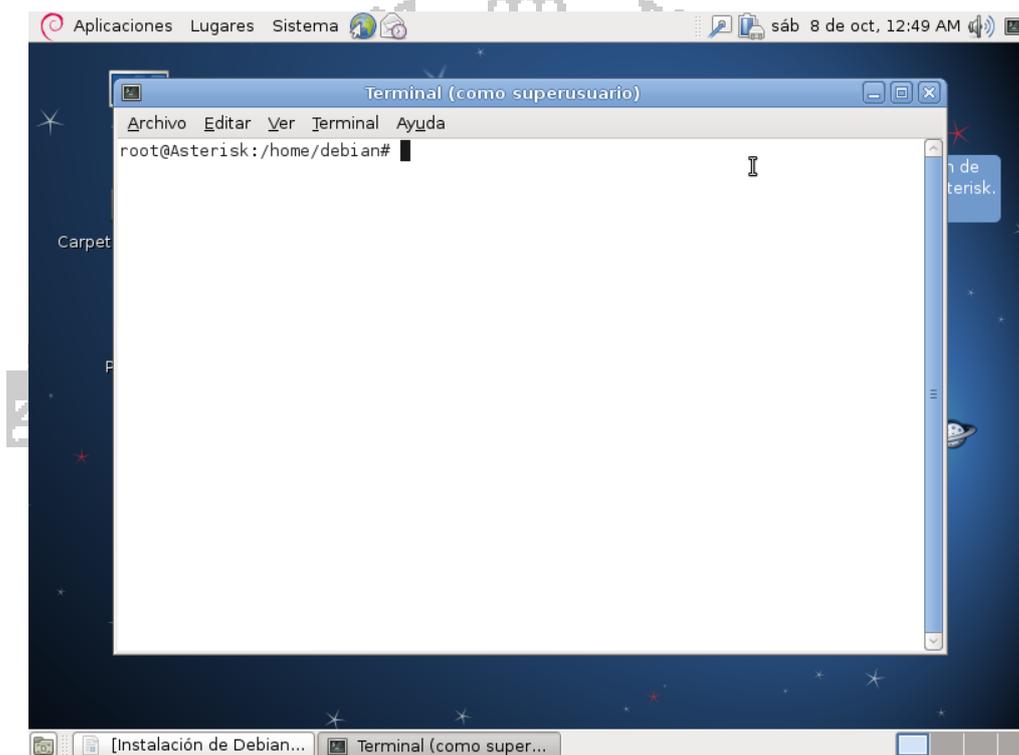
**Paso 22:** Ya tenemos instalado nuestro debían.



**Figura 62. Inicio del sistema operativo instalado**

#### 4.2.2. Instalación de Asterisk

Para la realización de los casos prácticos de los que está compuesto el proyecto se hará uso de la centralita Asterisk 10, cuya instalación y configuración se detalla a continuación, pero antes debemos preparar el entorno de nuestro sistema operativo, abriendo una terminal como superusuario en aplicaciones, como se muestra en el siguiente gráfico.



**Figura 63. Ejecutando terminal como superusuario**

Hecho esto de ahora en adelante siga los siguientes pasos:

**Paso 01:** Enlazamos a un servidor de USA en *nano /etc/apt/sources.list*.

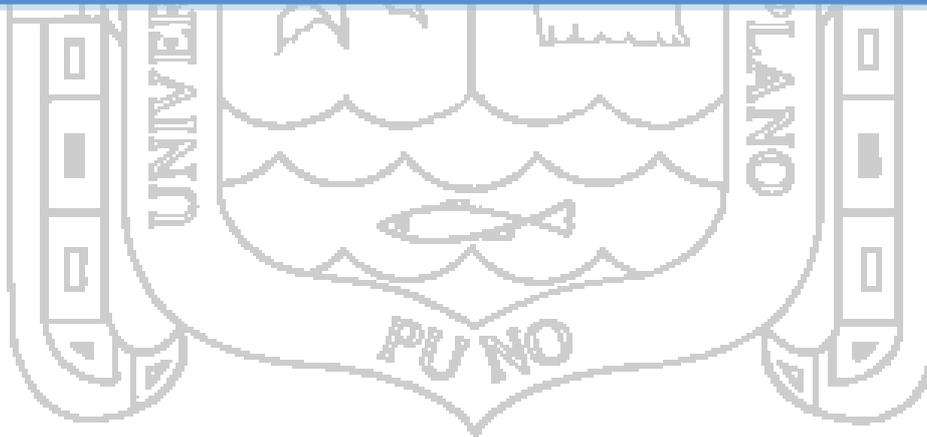
Como todos sabemos, en una distribución Linux, los repositorios de software son fundamentales, pues son ellos los que nos surten de paquetería en forma de software, y hablando de Debian, es una de las que más paquetería cuenta en su repositorio oficial, prácticamente no hay que hacer uso de repositorios de terceros para tener lo que queramos.

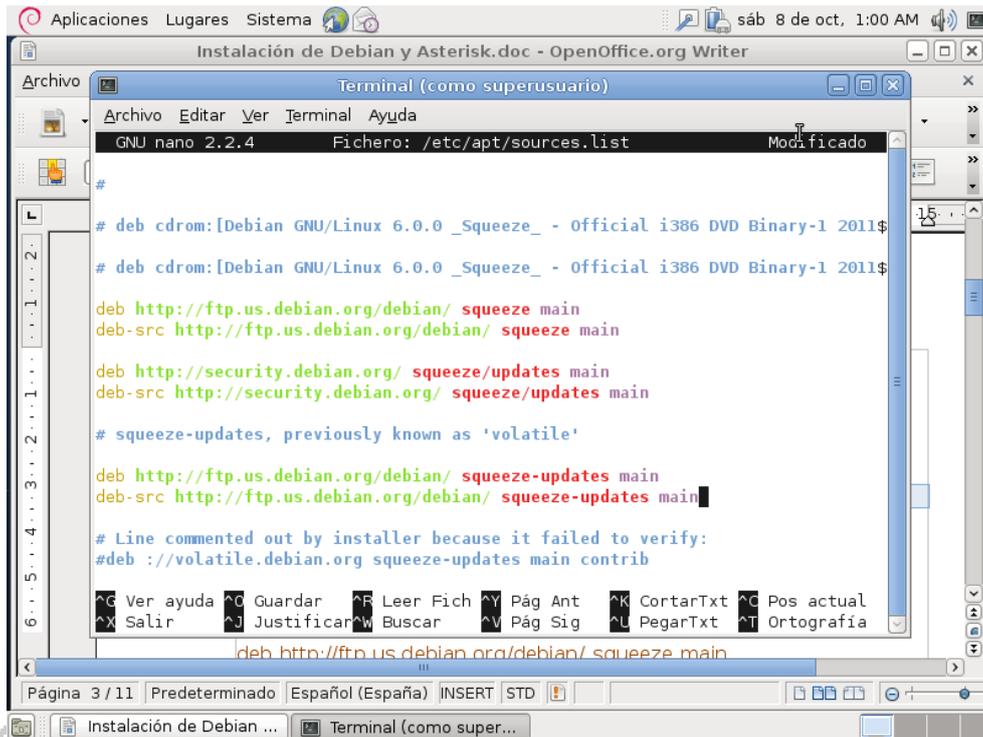
El gráfico siguiente muestra una terminal del fichero *sources.list*, al ejecutar en la terminal el comando como root:

```
# nano /etc/apt/sources.list
```

y copiando lo siguiente, borrando antes todo el contenido anterior del documento:

```
# deb cdrom:[Debian GNU/Linux 6.0.2.1 _Squeeze_ - Official i386 CD Binary-1 2011$  
# deb cdrom:[Debian GNU/Linux 6.0.2.1 _Squeeze_ - Official i386 CD Binary-1 20110$  
  
deb http://ftp.us.debian.org/debian/ squeeze main  
deb-src http://ftp.us.debian.org/debian/ squeeze main  
  
deb http://security.debian.org/ squeeze/updates main  
deb-src http://security.debian.org/ squeeze/updates main  
  
# squeeze-updates, previously known as 'volatile'  
  
deb http://ftp.us.debian.org/debian/ squeeze-updates main  
deb-src http://ftp.us.debian.org/debian/ squeeze-updates main
```





```

GNU nano 2.2.4 Fichero: /etc/apt/sources.list Modificado
#
# deb cdrom:[Debian GNU/Linux 6.0.0 _Squeeze_ - Official i386 DVD Binary-1 2011$
# deb cdrom:[Debian GNU/Linux 6.0.0 _Squeeze_ - Official i386 DVD Binary-1 2011$
deb http://ftp.us.debian.org/debian/ squeeze main
deb-src http://ftp.us.debian.org/debian/ squeeze main
deb http://security.debian.org/ squeeze/updates main
deb-src http://security.debian.org/ squeeze/updates main
# squeeze-updates, previously known as 'volatile'
deb http://ftp.us.debian.org/debian/ squeeze-updates main
deb-src http://ftp.us.debian.org/debian/ squeeze-updates main
# Line commented out by installer because it failed to verify:
#deb ://volatile.debian.org squeeze-updates main contrib

```

**Figura 64. Configurando el fichero sources.list**

**Paso 02:** Actualizaremos nuestro índice de los repositorios de nuestro Debian:

```

aptitude update
aptitude upgrade

```

**Paso 03:** Guardamos y reiniciamos:

```

reboot

```

nota: si da un error de código 4 fsck, para solucionarlo escribimos el comando:

```

fsck -c

```

**Paso 04:** Instalamos paquetes opcionales:

```
aptitude install ssh ntp
```

El paso siguiente será la descarga de la centralita de Asterisk y de los módulos DAHDI y libPRI.

La página oficial de Asterisk <http://www.asterisk.org>, contiene todo lo necesario para poder instalar correctamente la centralita.

**Paso 05:** Instalamos dependencias necesarias:

```
aptitude install build-essential libxml2-dev ncurses-dev libcurl3-dev libssl-dev  
libvorbis-dev libspeex-dev unixodbc unixodbc-dev libiksemel-dev flex xsltproc  
wget bzip2  
aptitude install linux-headers-`uname -r` g++ libncurses5-dev libnewt-dev libusb-  
dev subversion git-core gawk sudo libstdc++6
```

**Paso 06:** Descarga y descompresión de paquetes:

```
cd /usr/src  
wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-10-  
current.tar.gz  
wget http://downloads.asterisk.org/pub/telephony/DAHDI-linux-complete/DAHDI-  
linux-complete-2.6.0-rc1+2.6.0-rc1.tar.gz  
wget http://downloads.asterisk.org/pub/telephony/libpri/libpri-1.4-current.tar.gz  
wget http://downloads.asterisk.org/pub/telephony/libss7/releases/libss7-  
1.0.2.tar.gz
```

Comenzamos descomprimiendo todo:

```
tar xvzf asterisk-10-current.tar.gz && tar xvzf dahdi-linux-complete-2.6.0-rc1+2.6.0-rc1.tar.gz && tar xvzf libpri-1.4-current.tar.gz && tar xvzf libss7-1.0.2.tar.gz
```

Veamos lo que tenemos:

```
ls  
rm -R *.gz  
ls
```

**Paso 07:** Instalamos DAHDI:

```
cd /usr/src/dahdi-linux-complete-2.6*  
make  
make install  
make config
```

**Paso 08:** Instalamos Libpri:

```
cd /usr/src/libpri-1.4*  
make && make install
```

**Paso 09:** Instalamos libSS7:

```
cd /usr/src/libss7-1.0*  
make  
make install
```

**Paso 10:** Instalamos Asterisk:

Ahora vamos a instalar Asterisk. Para disponer de todas las características necesarias, vamos a hacer uso de uno de los scripts, no muy documentados, pero que por suerte conocemos y vamos a poner en práctica para facilitarnos la vida considerablemente.

```
cd /usr/src/asterisk-10*
```

Vamos a instalar todos los paquetes necesarios si se encuentran en nuestra distribución.

```
cd contrib/scripts
```

Si hay algún paquete que no viene con nuestra distribución, así lo instalamos a mano.

```
./install_prereq install  
./install_prereq install-unpackaged
```

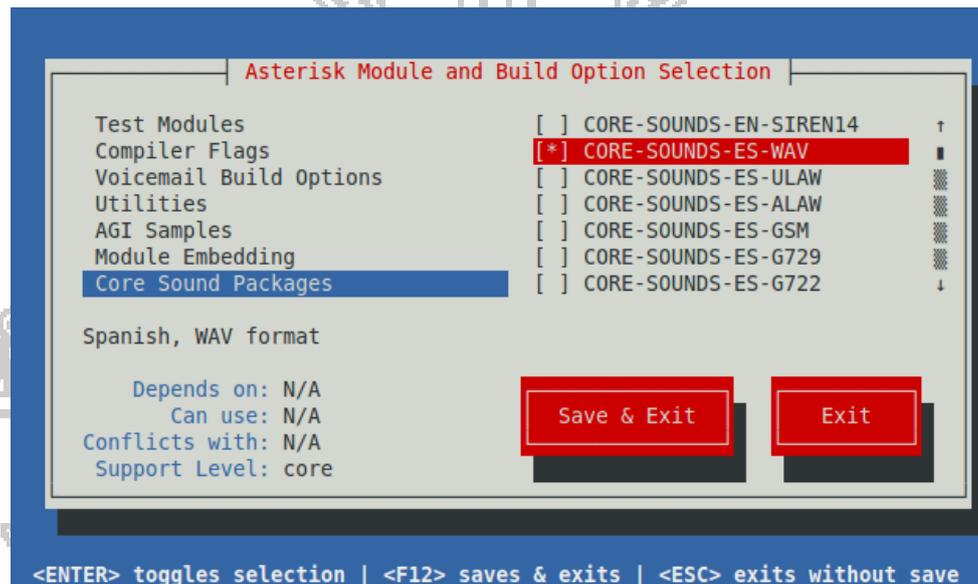
Con estos comandos, habremos instalado todas las dependencias necesarias para poder utilizar absolutamente todas las características de Asterisk (olvidándonos de conocer qué paquetes hacen falta para cada característica). Cuando nos pregunten por el código de la región, en la configuración del paquete libvpb0, elegiremos 51 si estamos en Perú.

A continuación, vamos a acceder al directorio del código fuente de Asterisk y preparar la compilación:

Con el siguiente comando compilamos y ejecutamos el menú de configuración de módulos:

```
make menuselect
```

Se abrirá la ventana de selección de componentes. En la sección seleccionaremos Core Sound Packages, seleccionaremos -SOUNDS-ES-WAV, y también CORE-SOUNDS-EN-WAV



**Figura 65. Configuración de Core Sound Packages**

El resto de opciones las dejaremos tal cual, y le daremos a “Save & Exit”.

Una vez seleccionado todo lo que queremos compilar, ya sólo queda la última parte que tardará dependiendo de la velocidad de la CPU.

```
make
make install
make samples
```

Copiamos los archivos de inicio para que el servicio se inicie automáticamente:

```
make config  
asterisk -vvvc  
stop now
```

**Paso 11:** Descarga e instalación las voces en español de voipnovatos:

El juego completo de voces consta de dos paquetes: “core-sounds” y “extra-sounds”. Descargaré los sonidos en `/var/lib/asterisk/sounds/es/`:

```
cd /var/lib/asterisk/sounds/es/  
  
wget http://www.voipnovatos.es/voces/voipnovatos-core-sounds-es-ulaw-  
1.4.tar.gz  
wget http://www.voipnovatos.es/voces/voipnovatos-extra-sounds-es-ulaw-  
1.4.tar.gz
```

Con esto obtenemos los archivos necesarios para disponer de las voces en español.

Ahora pasamos a descomprimir los archivos:

```
tar zxvf voipnovatos-core-sounds-es-ulaw-1.4.tar.gz  
tar zxvf voipnovatos-extra-sounds-es-ulaw-1.4.tar.gz
```

Una vez descomprimidos, vamos a enviarlos a su ubicación correcta, para que puedan ser usados por nuestro sistema:

```
cd ../  
mv es/dictate/es/* es/dictate/  
mv es/letters/es/* es/letters/  
mv es/silence/es/* es/silence/  
mv es/followme/es/* es/followme/  
mv es/phonetic/es/* es/phonetic/  
mv es/digits/es/* es/digits/  
mv es/es/* es/  
rm -rf es/es
```

Por último, para que Asterisk pueda encontrar los ficheros de sonido en el directorio “es” dentro de `/var/lib/asterisk/sounds/` debemos asegurarnos que en el fichero `/etc/asterisk/asterisk.conf` se encuentra la siguiente línea sin comentar:

```
languageprefix = yes
```

También habrá que especificar el lenguaje en la configuración de los canales en los ficheros `/etc/asterisk/sip.conf` y `/etc/asterisk/extensions.conf`. Normalmente utilizando la opción `language`:

```
language = es
```

Una vez hayamos hecho esto tendremos una centralita Asterisk funcionando y lista para ser configurada.

#### 4.2.3. Configuración de Asterisk

El objetivo es implementar, en una red local a modo de prueba, usando el protocolo SIP y IAX en DUNDi para interconectar dos o más servidores

Asterisk, logrando establecer llamadas entre ellos a través de las extensiones locales descritas en sus planes de marcación correspondientes.

#### 4.2.3.1. Configurando el protocolo DUNDi

Cuando hay que conectar un número considerable de servidores Asterisk y compartir las rutas y las extensiones configuradas en cada uno de ellos, la solución más funcional es el protocolo DUNDi. DUNDi (Distributed Universal Number Discovery) es un protocolo que permite buscar y compartir plan de llamadas entre servidores Asterisk. Esto por medio de una red Peer-to-peer (punto-a-punto) en la cual no existen roles fijos de clientes y servidores, sino que los servidores pueden asumir uno u otro rol.

En este ejemplo vamos a interconectar tres servidores Asterisk A, B y C. con los siguientes datos.

##### Servidor A:

IP: 192.168.1.40

MAC: 14:fe:b5:b6:45:74

##### Servidor B:

IP: 192.168.1.50

MAC: 08:00:27:1a:2e:e9

##### Servidor C:

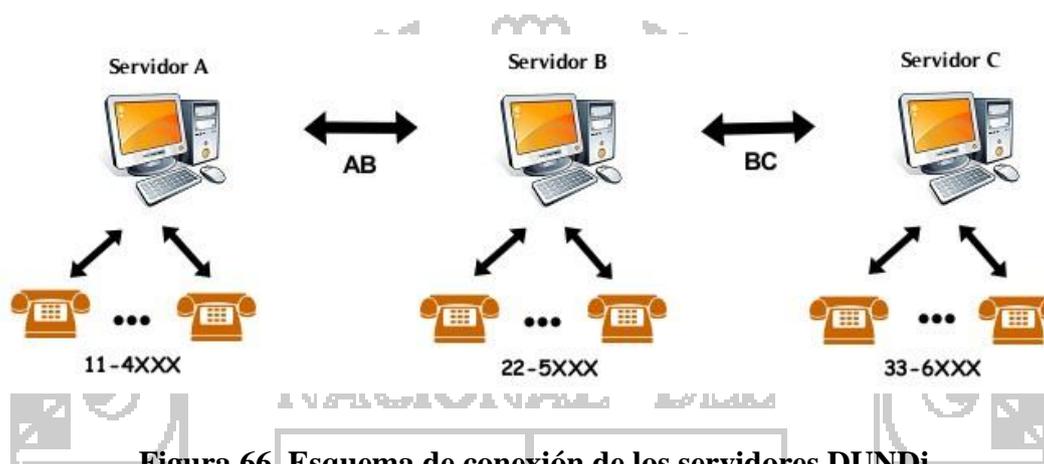
IP: 192.168.1.60

MAC: 08:00:27:a8:f1:2e

A lo largo de la configuración se puede decidir si compartir contextos ya configurados en el Dialplan (plan de llamadas) o crear contextos nuevos donde se indican las rutas que se quieren compartir. Aunque la configuración y puesta en marcha pueda parecer algo complicado, con un poco de práctica se irán aclarando las ideas y los conceptos claves.

Cada servidor gestiona las comunicaciones de una red local formada por tres terminales definidos en cada uno de ellos, y con una numeración distinta en función al servidor al que pertenezca.

Será necesario crear un plan de marcación para cada uno y configurar los archivos *iax.conf*, *extensions.conf*, *dundi.conf*, *iax.conf*, e *voicemail.conf*, así como crear unas llaves para la comunicación codificada.



**Figura 66. Esquema de conexión de los servidores DUNDi**

El servidor A tiene dentro de su red local cuatro extensiones, 4001, 4002, 4003 y 4004, las cuales se comunican internamente sin necesidad de un prefijo, sin embargo, para poder contactar con ellas desde otro servidor habrá que marcar 11 + n° local.

Los servidores B y C funcionan de la misma forma, tienen cuatro extensiones cada uno, desde la 5001 a la 5004 en el servidor B y desde la 6001 a la 6004 en el servidor C.

Para llamar a ellas desde otro servidor se hará mediante el prefijo correspondiente, 22 para las extensiones que estén definidas en el servidor B, y 33 para las que lo estén en el servidor C.

La comunicación entre el servidor A y el servidor B se llevará a cabo gracias al recurso DUNDi AB que se definirá posteriormente en el archivo *dundi.conf*. Así mismo, la comunicación entre los servidores B y C será gracias al recurso DUNDi BC definido de igual forma en el archivo *dundi.conf*.

**Fichero iax.conf:**

La configuración para los servidores A, B y C en *nano /etc/asterisk/iax.conf* contendrá los siguientes parámetros:

`type=friend` : Especifica el tipo de comunicación, en este caso bidireccional.

`dbsecret=dundi/secret` : Contraseña que se usará para enviar información a otro servidor.

`context=entrantesdundi` : Contexto dónde serán redirigidas las llamadas entrantes de otros servidores.

```
[truncal]
type=friend
dbsecret=dundi/secret
context=entrantesdundi
```

**Llaves de autenticación:**

Las llaves de autenticación permiten tener acceso a los demás servidores por medio de un enlace troncal iax. Al crearlas en el directorio */var/lib/asterisk/keys*, se generan dos llaves con las extensiones *pub* y *key*, las cuales se compartirán con el servidor más próximo.

En el servidor A se crearán unas llaves llamadas *llaveAB.pub* y *llaveAB.key*, las cuáles se copiarán al servidor B en el directorio */var/lib/asterisk/keys*, con el siguiente comando:

```
root@Bruno:~# cd /var/lib/asterisk/keys
root@Bruno:/var/lib/asterisk/keys# astgenkey -n llaveAB
```

This script generates an RSA private and public key pair in PEM format for use by Asterisk. You will be asked to enter a passcode for your key multiple times. Please enter the same code each time. The resulting files will need to be moved to /var/lib/asterisk/keys if you want to use them, and any private keys (.key files) will need to be initialized at runtime either by running Asterisk with the '-i' option, or with the 'init keys' command once Asterisk is running.

Press ENTER to continue or ^C to cancel.

Generating SSL key 'llaveAB':

Generating RSA private key, 1024 bit long modulus

..++++++

.....++++++

e is 65537 (0x10001)

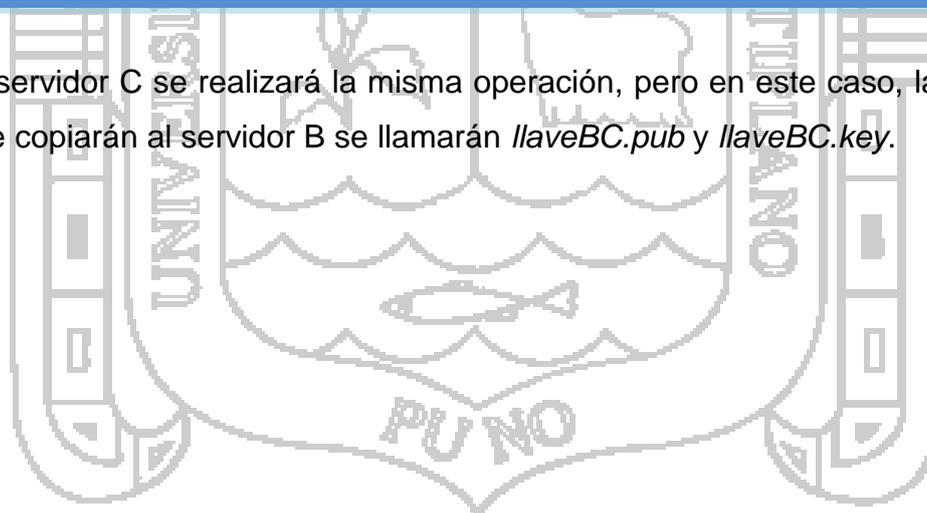
writing RSA key

Key creation successful.

Public key: llaveAB.pub

Private key: llaveAB.key

En el servidor C se realizará la misma operación, pero en este caso, las llaves que se copiarán al servidor B se llamarán *llaveBC.pub* y *llaveBC.key*.



```
root@Asterisk:/etc/asterisk# cd /var/lib/asterisk/keys
root@Asterisk:/var/lib/asterisk/keys# astgenkey -n llaveBC
```

This script generates an RSA private and public key pair in PEM format for use by Asterisk. You will be asked to enter a passcode for your key multiple times. Please enter the same code each time. The resulting files will need to be moved to /var/lib/asterisk/keys if you want to use them, and any private keys (.key files) will need to be initialized at runtime either by running Asterisk with the '-i' option, or with the 'init keys' command once Asterisk is running.

Press ENTER to continue or ^C to cancel.

```
Generating SSL key 'llaveBC':
Generating RSA private key, 1024 bit long modulus
.....++++++
.....++++++
e is 65537 (0x10001)
writing RSA key
Key creation successful.
Public key: llaveBC.pub
Private key: llaveBC.key
```

Por lo tanto, el servidor A contendrá las llaves “llaveAB”, el servidor B las llaves “llaveAB” y “llaveBC” y el servidor C las “llaveBC”. Estando todos comunicados entre sí gracias al servidor B.

#### **Fichero: extensions.conf**

Al igual que en el anterior fichero, en la línea de comandos escribimos *nano /etc/asterisk/extensions.conf*. Una vez dentro el archivo de configuración contendrá los siguientes contextos:

[general] : Define las características comunes para todos los contextos.

[buscardundi] : Contexto último para encontrar la extensión solicitada en otros servidores por medio del recurso DUNDi definido en dundi.conf.

[compartir] : Contexto dónde se publican las extensiones que se comparten con otros servidores, con el formato: prefijo + nº local.

[entrantesdundi] : Contexto al que llegan las llamadas entrantes de otros servidores, dichas llamadas será redirigidas al contexto local correspondiente.

[local] : Este contexto (o contextos) incluye a todas las extensiones del plan de marcación.



**Servidor A:**

```
[general]
static=yes
writeprotect=no
autofallthrough=no
clearglobalvars=yes
priorityjumping=no

[buscardundi]
switch => DUNDi/AB
switch => DUNDi/BC

[compartir]
exten => _114XXX,1,NoOp

[entrantesdundi]
exten => _114XXX,1,Goto(local-a,${EXTEN:2},1)

[local-a]

include => buscardundi

exten => _4xxx,1,Dial(SIP/${EXTEN},15,Ttm)
exten => _4xxx,2,Voicemail(${EXTEN}@voicemail)
exten => _4xxx,3,Playback(vm-goodbye)
exten => _4xxx,4,Hangup

exten => *98,1,Answer
exten => *98,2,Wait(1)
exten => *98,3,VoiceMailMain(${CALLERID}@voicemail)
exten => *98,4,Hangup

exten => _5xxx,1,Dial(SIP/${EXTEN}@AsteriskB,10)
exten => _5xxx,2,Voicemail(${EXTEN}@voicemail)
exten => _5xxx,3,Playback(vm-goodbye)
exten => _5xxx,n,Hangup()
```

**Servidor B:**

```
[general]
static=yes
writeprotect=no
autofallthrough=no
clearglobalvars=yes
priorityjumping=no

[buscardundi]
switch => DUNDi/AB
;switch => DUNDi/BC

[compartir]
exten => _225XXX,1,NoOp

[entrantesdundi]
exten => _225XXX,1,Goto(local-b,${EXTEN:2},1)

[local-b]

include => buscardundi

exten => _5xxx,1,Dial(SIP/${EXTEN},15,Ttm)
exten => _5xxx,2,Voicemail(${EXTEN}@voicemail)
exten => _5xxx,3,Playback(vm-goodbye)
exten => _5xxx,4,Hangup

exten => *98,1,Answer
exten => *98,2,Wait(1)
exten => *98,3,VoiceMailMain(${CALLERID}@voicemail)
exten => *98,4,Hangup
```



**Servidor C:**

```
[general]
static=yes
writeprotect=no
autofallthrough=no
clearglobalvars=yes
priorityjumping=no

[buscardundi]
switch => DUNDi/AB
switch => DUNDi/BC

[compartir]
exten => _336XXX,1,NoOp

[entrantesdundi]

exten => _336XXX,1,Goto(local-c,${EXTEN:2},1)

[local-c]

include => buscardundi

exten => _6xxx,1,Dial(SIP/${EXTEN},15,Ttm)
exten => _6xxx,2,Voicemail(${EXTEN}@voicemail)
exten => _6xxx,3,Playback(vm-goodbye)
exten => _6xxx,4,Hangup

exten => *98,1,Answer
exten => *98,2,Wait(1)
exten => *98,3,VoiceMailMain(${CALLERID}@voicemail)
exten => *98,4,Hangup
```

**Fichero: dundi.conf**

Este archivo de configuración definirá los parámetros y recursos necesarios para que la comunicación se lleve a cabo usando el protocolo DUNDi

Se usarán como direcciones MAC y direcciones IP de los servidores A, B y C

## Servidor A

Nuestra primera parada es el fichero del servidor A en *nano* */etc/asterisk/dundi.conf* . Vamos por partes:

```
[general]
department=IT
organization=UPCT
locality=CT
stateprov=MU
country=SP
email=prueba@voip.com
phone=666666666
```

En esta primera parte consiste en indicar quiénes somos y la información de contacto. Esto permitirá a otros nodos de la red conocernos.

```
bindaddr=0.0.0.0
port=4520
```

En *bindaddr* definimos la dirección IP de conexión con el servidor. En este caso escucha en cualquier interfaz, y con *port* definimos el puerto DUNDi por defecto.

```
entityid=14:fe:b5:b6:45:74
```

El *entityid* es un parámetro importante. Es nuestra identificación en la red, y debe ser única. Si no lo definimos, se usará la dirección MAC de la primera interfaz de red, pero para más seguridad conviene definirlo.

```
cachetime=5
ttl=3
autokill=yes
```

El tiempo máximo de búsqueda de respuesta por defecto es 3600 segundos, al poner cachetime=5 limitamos su búsqueda. En ttl (time-to-live), no necesitaremos conocer “personalmente” a todos los nodos de la red. Nuestros vecinos pueden propagar nuestras consultas a los suyos, y así sucesivamente hasta que el ttl llegue a 0. Poniendo ttl=3 limitaremos la profundidad de las consultas, reduciendo el tiempo de espera. En caso de no obtener respuesta de una extensión, autokill de Asterisk envía CANCEL a los peers.

```
[mappings]
AB=>compartir,0,IAX2,troncal:${SECRET}@192.168.1.40/${NUMBER},nopartial
```

Esta sección es clave. Por un lado definimos los recursos que vamos a usar, por otro indicaremos los números que nuestra centralita publicará. Vamos a detenernos y a analizar cada uno de estos elementos.

- AB : Este es el nombre del recurso. Lo usaremos solamente para buscar extensiones en la empresa, en las diferentes sedes.
- compartir : Es el contexto donde tenemos definidas nuestras extensiones. Cuando otro nodo busque una extensión que tenemos definida en este contexto, responderemos con gusto.
- 0 : Es el peso de nuestra respuesta. Cuando menor sea más peso (prioridad). Es útil en otro tipo de recursos. A la hora de buscar rutas de menor coste para las llamadas, si estamos seguros de que somos la mejor, pondremos 0. Si nuestra ruta es buena, pero las hay mejor, pondremos un valor mayor.
- IAX2 : Protocolo usado para la comunicación DUNDi. Puede ser SIP, H323 o cualquier otro.

troncal : En una cadena de llamada IAX2, “troncal” enlace definido en el archivo iax.conf. Es un usuario para la comunicación de los servidores.

`#{SECRET}` Variable que contiene la clave del recurso DUNDi, se sustituirá por la contraseña a utilizar para autenticar el enlace troncal.

192.168.1.40 : Es dirección IP del servidor A.

`#{NUMBER}` : Sustituirá por el número de la consulta.

nopartial : Opción usada para prevenir que se entreguen resultados que contengan sólo una parte de la extensión.

```
[08:00:27:1a:2e:e9]
model=symmetric
host=192.168.1.50
inkey=llaveAB
outkey=llaveAB
include=all
permit=all
qualify=yes
order=primary
```

`[08:00:27:1a:2e:e9]` : Identificador de red con el que se comparte el recurso DUNDi con el Servidor B (puede ser la dirección MAC).

`model=symmetric` : Permite hacer y recibir consultas.

`host=bbb.bbb.bbb.bbb` : Dirección IP del servidor B.

`inkey=llaveAB` : Nombre de la llave usada para recibir la consulta.

`outkey=llaveAB` : Nombre de la llave usada para realizar la consulta.

- `include=all` : Nombre de aquello que queremos compartir con el servidor. Es posible usar `all` para permitir todo el mapeo.
- `permit=all` : Contextos DUNDi a los que tendrá acceso este nodo.
- `qualify=yes` : Se controla periódicamente que la conexión con el nodo esté activa.
- `order=primary` : Orden de búsqueda para cada servidor. Puede ser `primary`, `secondary`, `tertiary` y `quartary`.



## Servidor B

```
[general]
department=IT
organization=UPCT
locality=CT
stateprov=MU
country=SP
email=prueba@voip.com
phone=666666666
bindaddr=0.0.0.0
port=4520
entityid=08:00:27:1a:2e:e9
cachetime=5
ttl=3
autokill=yes

[mappings]
AB=>compartir,0,IAX2,troncal:${SECRET}@192.168.1.50/${NUMBER},nopartial
BC=>compartir,0,IAX2,troncal:${SECRET}@192.168.1.50/${NUMBER},nopartial

[14:fe:b5:b6:45:74]
model=symmetric
host=192.168.1.40
inkey=llaveAB
outkey=llaveAB
include=all
permit=all
qualify=yes
order=primary

[08:00:27:a8:f1:2e]
model=symmetric
host=192.168.1.60
inkey=llaveBC
outkey=llaveBC
include=all
permit=all
qualify=yes
order=primary
```

## Servidor C

```
[general]
department=IT
organization=UPCT
locality=CT
stateprov=MU
country=SP
email=prueba@voip.com
phone=666666666
bindaddr=0.0.0.0
port=4520
entityid=08:00:27:a8:f1:2e
cachetime=5
ttl=3
autokill=yes

[mappings]
BC=>compartir,0,IAX2,troncal:${SECRET}@192.168.1.60/${NUMBER},nopartial

[08:00:27:1a:2e:e9]
model=symmetric
host=192.168.1.50
inkey=llaveBC
outkey=llaveBC
include=all
permit=all
qualify=yes
order=primary
```

### Fichero: sip.conf

El archivo *sip.conf* contiene parámetros relacionados con la configuración SIP de Asterisk, más adelante se configurará como si de otro contexto se tratase. Por ahora en este archivo se definirán las extensiones locales para cada servidor y su contexto correspondiente.

## Servidor A

```
[general]
bindport=4569
bindaddr=0.0.0.0
delayreject=yes
disallow=all
allow=gsm
allow=ulaw
allow=alaw
autokill=yes
```

[general] :	En primer lugar existe la sección [general], donde se definen variables globales y aspectos por defecto para todos los canales SIP.
bindport=4569 :	El puerto utilizado para conectar las extensiones SIP al servidor Asterisk (protocolo UDP).
bindaddr=0.0.0.0 :	Interfaz de red que utilizará Asterisk para atender las conexiones SIP entrantes. El valor por defecto es 0.0.0.0, es decir, escucha en todas las interfaces de red disponibles.
delayreject=yes :	Si se recibe una contraseña incorrecta en un canal IAX, esto retrasará el envío del REGREQ o AUTHREP rechazar los mensajes, lo que ayudará a asegurar contra ataques de contraseña de fuerza bruta. El tiempo de retraso es de 1.000 ms.
disallow=all :	Deshabilitar todos los codecs (audio y video)
allow=gsm :	Permite el códec gsm.
allow=ulaw :	permite el códec g711u.
allow=alaw :	permite el códec g711a.

autokill=yes :                    En caso de no obtener respuesta de una extensión,  
autokill envía CANCEL a los peers.

En esta parte se definen tanto los clientes que se conectarán al servidor Asterisk A. Para fines prácticos solo se enumera cuatro extensiones:

```
[4001]
type=friend
username=4001
secret=pass
host=dynamic
callerid="Bruno" <4001>
context=local-a
mailbox=4001@voicemail
qualify=yes

[4002]
type=friend
username=4002
secret=pass
host=dynamic
callerid="Fredy" <4002>
context=local-a
mailbox=4002@voicemail
qualify=yes

[4003]
type=friend
username=4003
secret=pass
host=dynamic
callerid="Sony" <4003>
context=local-a
mailbox=4003@voicemail
qualify=yes

[4004]
type=friend
username=4004
secret=4004
host=dynamic
callerid="Linphone" <4003>
context=local-a
mailbox=4004@voicemail
qualify=yes
```

## Servidor B

```
[general]
bindport=4569
bindaddr=0.0.0.0
delayreject=yes
disallow=all
allow=gsm
allow=ulaw
allow=alaw
autokill=yes

[5001]
type=friend
username=5001
secret=pass
host=dynamic
callerid="Fredy" <5001>
context=local-b
mailbox=5001@voicemail
qualify=yes

[5002]
type=friend
username=5002
secret=pass
host=dynamic
callerid="Joel" <5002>
context=local-b
mailbox=5002@voicemail
qualify=yes

[5003]
type=friend
username=5003
secret=pass
host=dynamic
callerid="Nery" <5003>
context=local-b
mailbox=5003@voicemail
qualify=yes
```

## Servidor C

```
[general]
bindport=4569
bindaddr=0.0.0.0
delayreject=yes
disallow=all
allow=gsm
allow=ulaw
allow=alaw
autokill=yes

[6001]
type=friend
username=6001
secret=6001
host=dynamic
callerid="Tania" <6001>
context=local-c
mailbox=6001@voicemail
qualify=yes

[6002]
type=friend
username=6002
secret=6002
host=dynamic
callerid="Nery" <6002>
context=local-c
mailbox=6002@voicemail
qualify=yes

[6003]
type=friend
username=6003
secret=6003
host=dynamic
callerid="Genoveva" <6003>
context=local-c
mailbox=6003@voicemail
qualify=yes
```

### Fichero: voicemail.conf

El archivo *voicemail.conf* contiene la definición de todas las prestaciones del buzón de voz. Para entrar a la plantilla de configuración, en la línea de comandos tecleamos *nano /etc/asterisk/voicemail.conf*. Solo deben agregar los siguientes datos:

#### Servidor A

```
[voicemail]
4001 => 4001,Bruno,bruno@localhost
4002 => 4002,Fredy,fredy@localhost
4003 => 4003,Sony,sony@localhost
4004 => 4004,Linphone,linphone@localhost
```

#### Servidor B

```
[voicemail]
5001 => 5001,Fredy,fredy@localhost
5002 => 5002,Joel,joel@localhost
5003 => 5003,Nery,nery@localhost
```

#### Servidor C

```
[voicemail]
6001 => 6001,Tania,tania@localhost
6002 => 6002,Nery,nery@localhost
6003 => 6003,Genoveva,genoveva@localhost
6004 => 6004,Isabel,isabel@localhost
```

Entre las distintas comprobaciones del protocolo DUNDi que se pueden realizar en la línea de comandos, nos encontramos con:

```
CLI > dundi show peers
```

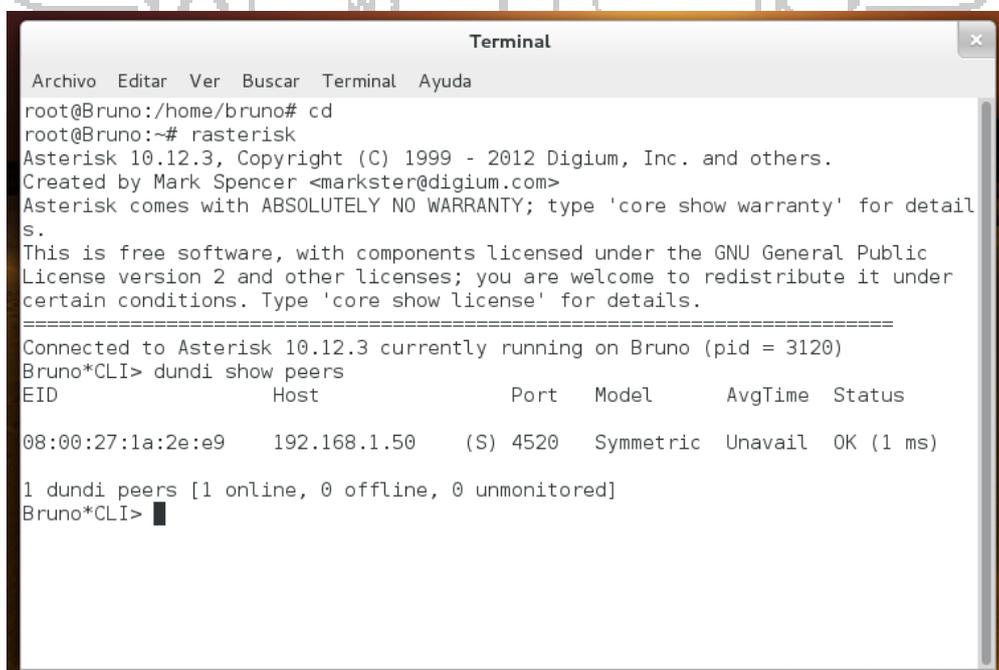
La cuál mostrará la dirección MAC y la IP del servidor dónde se ejecuta, y el estado actual. Sin más preámbulo comprobemos el funcionamiento de DUNDi.

### Trunk DUNDi Local Asterisk

Como se ve en las imágenes siguientes, el Trunk<sup>2</sup> DUNDi enlaza tres servidores de la misma red y aplicando en la línea de comandos los siguientes códigos:

```
cd
rasterisk
reload
```

Se observa, la comprobación del registro de la siguiente manera: En el servidor A se registra el servidor B, en el servidor B se registran los servidores A y C, y por último, en el servidor C se registra el B, de esta forma queda comprobada la comunicación entre los tres servidores.



```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
root@Bruno:/home/bruno# cd
root@Bruno:~# rasterisk
Asterisk 10.12.3, Copyright (C) 1999 - 2012 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public License version 2 and other licenses; you are welcome to redistribute it under certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 10.12.3 currently running on Bruno (pid = 3120)
Bruno*CLI> dundi show peers
EID                Host                Port  Model    AvgTime  Status
08:00:27:1a:2e:e9  192.168.1.50        (S) 4520  Symmetric Unavail  OK (1 ms)

1 dundi peers [1 online, 0 offline, 0 unmonitored]
Bruno*CLI> █
```

**Figura 67. Comprobación del registro DUNDi en el servidor A**

<sup>2</sup> Un Trunk es una conexión, bien entre servidores o entre un servidor y las aplicaciones de un operador de Telefonía IP que permite el tráfico de VoIP.

```

Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
root@Bruno:/home/brunofchm# cd
root@Bruno:~# rasterisk
Asterisk 10.12.3, Copyright (C) 1999 - 2012 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 10.12.3 currently running on Bruno (pid = 1356)
Bruno*CLI> dundi show peers
EID                Host                Port  Model    AvgTime  Status
08:00:27:a8:f1:2e  192.168.1.60        (S) 4520  Symmetric Unavail  OK (2 ms)
14:fe:b5:b6:45:74  192.168.1.40        (S) 4520  Symmetric Unavail  OK (1 ms)

2 dundi peers [2 online, 0 offline, 0 unmonitored]
Bruno*CLI>
    
```

**Figura 68. Comprobación del registro DUNDi en el servidor B**

```

Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
root@Asterisk:/home/fchm# cd
root@Asterisk:~# rasterisk
Privilege escalation protection disabled!
See https://wiki.asterisk.org/wiki/x/1gKfFAQ for more details.
Asterisk 10.12.4, Copyright (C) 1999 - 2012 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 10.12.4 currently running on Asterisk (pid = 1443)
Asterisk*CLI> dundi show peers
EID                Host                Port  Model    AvgTime  Status
08:00:27:1a:2e:e9  192.168.1.50        (S) 4520  Symmetric Unavail  OK (4 ms)

1 dundi peers [1 online, 0 offline, 0 unmonitored]
Asterisk*CLI>
    
```

**Figura 69. Comprobación del registro DUNDi en el servidor C**

#### 4.2.3.2. Configurando el protocolo SIP

En este escenario tenemos dos localidades que queremos integrar con el protocolo SIP, así que tendré dos equipos (Servidor A y Servidor B) con los siguientes datos:

##### Servidor A:

IP: 192.168.1.40

MAC: 14:fe:b5:b6:45:74

##### Servidor B:

IP: 192.168.1.50

MAC: 08:00:27:1a:2e:e9

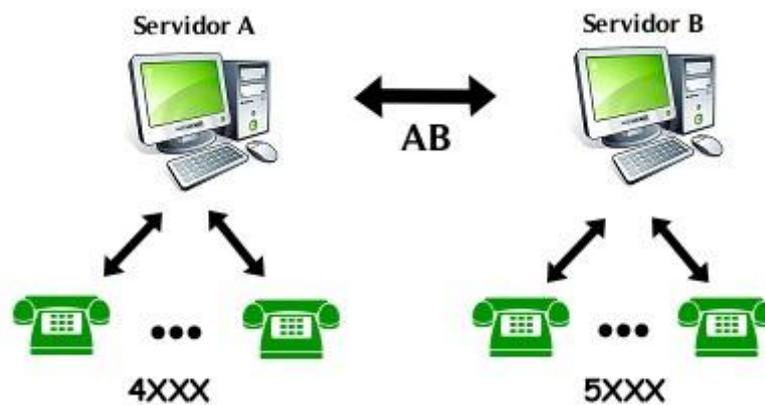


Figura 70. Esquema de conexión de los servidores SIP

Como en el anterior ejemplo mostrado, será necesario crear un plan de marcación para ambos servidores y configurar los archivos *extensions.conf*, *sip.conf*, e *voicemail.conf*.

**Fichero:** sip.conf

El archivo *sip.conf* sirve para configurar todo lo relacionado con el protocolo SIP<sup>3</sup> y añadir nuevos usuarios o conectar con proveedores SIP.

Su configuración y modo de empleo es un tanto diferente que en el que usamos anteriormente, abra que modificar register en el fichero. Empezamos con la configuración del servidor A.

### Servidor A

```
[general]
context=default          ; Default context for incoming calls

nat=yes
externip=192.168.1.40
;externhost=cumunibruno.org
localnet=192.168.1.0/255.255.255.0

externrefresh=180
```

Cuando actúa como servidor, envía las peticiones de registro de usuarios al registrar, o al proxy activo si no hay registrar configurado. Si está habilitado el gateway registra los dial-peers de tipo voice-port o group también.

Configura la dirección IP o el nombre de dominio del registrar SIP donde enviar los mensajes de registrar. Si no se configura ninguno, los mensajes son enviados al proxy activo. Si tampoco hay proxy, en modo servidor, el propio router actúa como registrar.

---

<sup>3</sup> SIP (Session Initiation Protocol o Protocolo de Inicialización de Sesiones) es un protocolo propuesto como estándar por la IETF (Internet Engineering Task Force o Fuerza de Trabajo en Ingeniería de Internet), descrito en el RFC 3261, para la iniciación, modificación y finalización de sesiones interactivas de usuario, en las cuales intervienen elementos de medios electrónicos, como pueden ser vídeo, voz y mensajería instantánea. Es uno de los varios protocolos de señalización para la tecnología de Voz sobre IP (VoIP o Voice over IP) y forma parte de la arquitectura IMS (IP Multimedia Subsystem o Subsistema multimedia por IP).

```
register => AsteriskA:1234@192.168.1.50/AsteriskB
```

```
[general]  
bindport=4569  
bindaddr=0.0.0.0  
delayreject=yes  
disallow=all  
allow=gsm  
allow=ulaw  
allow=alaw  
autokill=yes
```

```
[4001]  
type=friend  
username=4001  
secret=pass  
host=dynamic  
callerid="Bruno" <4001>  
context=local-a  
mailbox=4001@voicemail  
qualify=yes
```

```
[4002]  
type=friend  
username=4002  
secret=pass  
host=dynamic  
callerid="Fredy" <4002>  
context=local-a  
mailbox=4002@voicemail  
qualify=yes
```

```
[4003]
type=friend
username=4003
secret=pass
host=dynamic
callerid="Sony" <4003>
context=local-a
mailbox=4003@voicemail
qualify=yes
```

```
[AsteriskB]
type=friend
username=AsteriskB
secret=1234
host=dynamic
context=local-a
mailbox=AsteriskB@voicemail
qualify=yes
language=es
insecure=invite
disallow=all
allow=ulaw
allow=alaw
```

### Servidor B

```
[general]
context=default ; Default context for incoming calls
nat=yes
externip=192.168.1.50
;externhost=comunibruno.org
localnet=192.168.1.0/255.255.255.0
externrefresh=180
```

```
register => AsteriskB:1234@192.168.1.20/AsteriskA
```

```
[general]  
bindport=4569  
bindaddr=0.0.0.0  
delayreject=yes  
disallow=all  
allow=gsm  
allow=ulaw  
allow=alaw  
autokill=yes
```

```
[5001]  
type=friend  
username=5001  
secret=pass  
host=dynamic  
callerid="Fredy" <5001>  
context=local-b  
mailbox=5001@voicemail  
qualify=yes
```

```
[5002]  
type=friend  
username=5002  
secret=pass  
host=dynamic  
callerid="Joel" <5002>  
context=local-b  
mailbox=5002@voicemail  
qualify=yes
```

```
[5003]
type=friend
username=5003
secret=pass
host=dynamic
callerid="Nery" <5003>
context=local-b
mailbox=5003@voicemail
qualify=yes
```

```
[AsteriskA]
type=friend
username=AsteriskA
secret=1234
host=dynamic
context=local-b
mailbox=AsteriskA@voicemail
qualify=yes
language=es
insecure=invite
disallow=all
allow=ulaw
allow=alaw
```

### **Fichero: extensions.conf**

Al igual que en el anterior fichero, en la línea de comandos escribimos *nano /etc/asterisk/extensions.conf*. Una vez dentro el archivo de configuración contendrá los siguientes contextos:

[general] : Define las características comunes para todos los contextos.

[buscardundi] : Contexto último para encontrar la extensión solicitada en otros servidores por medio del recurso DUNDi definido en dundi.conf.

[compartir] : Contexto dónde se publican las extensiones que se comparten con otros servidores, con el formato: prefijo + nº local.

[entrantesdundi] : Contexto al que llegan las llamadas entrantes de otros servidores, dichas llamadas será redirigidas al contexto local correspondiente.

[local] : Este contexto (o contextos) incluye a todas las extensiones del plan de marcación.



**Servidor A:**

```
[local-a]

exten => _4xxx,1,Dial(SIP/${EXTEN},15,Ttm)
exten => _4xxx,2,Voicemail(${EXTEN}@voicemail)
exten => _4xxx,3,Playback(vm-goodbye)
exten => _4xxx,4,Hangup

exten => *98,1,Answer
exten => *98,2,Wait(1)
exten => *98,3,VoiceMailMain(${CALLERID}@voicemail)
exten => *98,4,Hangup

exten => _5xxx,1,Dial(SIP/${EXTEN}@AsteriskB,10)
exten => _5xxx,2,Voicemail(${EXTEN}@voicemail)
exten => _5xxx,3,Playback(vm-goodbye)
exten => _5xxx,n,Hangup()
```

**Servidor B:**

```
[local-b]

exten => _5xxx,1,Dial(SIP/${EXTEN},15,Ttm)
exten => _5xxx,2,Voicemail(${EXTEN}@voicemail)
exten => _5xxx,3,Playback(vm-goodbye)
exten => _5xxx,4,Hangup

exten => *98,1,Answer
exten => *98,2,Wait(1)
exten => *98,3,VoiceMailMain(${CALLERID}@voicemail)
exten => *98,4,Hangup

exten => _4xxx,1,Dial(SIP/${EXTEN}@AsteriskA,10)
exten => _4xxx,2,Voicemail(${EXTEN}@voicemail)
exten => _4xxx,3,Playback(vm-goodbye)
exten => _4xxx,4,Hangup()
```

### Fichero: voicemail.conf

Como un último paso en la línea de comandos, agregamos el siguiente código de configuración. Entramos al fichero con el siguiente comando *nano /etc/asterisk/voicemail.conf*.

### Servidor A

```
[voicemail]
4001 => 4001,Bruno,bruno@localhost
4002 => 4002,Fredy,fredy@localhost
4003 => 4003,Sony,sony@localhost
AsteriskB => AsteriskB,asteriskb,asteriskb@localhost
```

### Servidor B

```
[voicemail]
5001 => 5001,Fredy,fredy@localhost
5002 => 5002,Joel,joel@localhost
5003 => 5003,Nery,nery@localhost
AsteriskA => AsteriskA,asteriska,asteriska@localhost
```

### Trunk SIP Local Asterisk

Una vez configurado los archivos de Asterisk en las Centrales de Telefonía IP, se procede a verificar la comunicación entre ellas. Al igual que en el caso anterior, para iniciar Asterisk se ejecuta el comando en cada central de telefonía IP: En el servidor A se registra el servidor B y en el servidor B se registra el servidor A, además se observa sus extensiones registradas correspondientes a cada servidor, de esta forma queda comprobada la comunicación entre los tres servidores.

```

Terminal
Archivo Editar Ver Buscar Terminal Ayuda
Asterisk 10.12.3, Copyright (C) 1999 - 2012 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public License version 2 and other licenses; you are welcome to redistribute it under certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 10.12.3 currently running on Bruno (pid = 3120)
Bruno*CLI> sip show peers
Name/username          Host                               Dyn Forcerport
ACL Port      Status  Description
4001/4001      5060    OK (15 ms)  192.168.1.41      D   N
4002/4002      51620   OK (1 ms)   192.168.1.43      D   N
4003/4003      0        UNKNOWN     (Unspecified)    D   N
4004/4004      0        UNKNOWN     (Unspecified)    D   N
AsteriskB/AsteriskB  192.168.1.50      D   N
5060          OK (7 ms)
5 sip peers [Monitored: 3 online, 2 offline Unmonitored: 0 online, 0 offline]
Bruno*CLI>
    
```

**Figura 71. Comprobación del registro SIP en el servidor A**

```

Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
root@Bruno:/home/brunofchm# cd
root@Bruno:~# rasterisk
Asterisk 10.12.3, Copyright (C) 1999 - 2012 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public License version 2 and other licenses; you are welcome to redistribute it under certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 10.12.3 currently running on Bruno (pid = 1356)
Bruno*CLI> sip show peers
Name/username          Host                               Dyn Forcerport
ACL Port      Status  Description
5001/5001      5060    OK (21 ms)  192.168.1.51      D   N
5002/5002      53704   UNREACHABLE 192.168.1.53      D   N
5003/5003      0        UNKNOWN     (Unspecified)    D   N
AsteriskA/AsteriskA  192.168.1.40      D   N
5060          OK (5 ms)
4 sip peers [Monitored: 2 online, 2 offline Unmonitored: 0 online, 0 offline]
Bruno*CLI>
    
```

**Figura 72. Comprobación del registro SIP en el servidor B**

### Proceso de llamada entre extensiones registradas

Los procesos de llamar y esperar llamada son ejecutados a la par en las instancias SIP (teléfonos IP) involucrados en el establecimiento de llamada. Es decir un usuario del sistema de telefonía IP después de enrolar su número de extensión personal debe activar el servicio de esperar llamada con el fin de ser alcanzado por los otros usuarios en el sistema.

En las siguientes figuras se detallan el funcionamiento de ambos procesos. Para el presente proyecto contamos con teléfonos IP<sup>4</sup> LINKSYS SPA 841, como se muestra en la figura 73, y los SoftPhones<sup>5</sup> X-Lite CounterPatch, este último utilizaremos como ejemplo práctico.



**Figura 73. Teléfono IP LINKSYS SPA 841**

Una vez configurado el Username, Authorization y Password, los mismos creados en el archivo *sip.conf*, para el servidor A con la extensión 4002 y para

<sup>4</sup> Un teléfono de VoIP o teléfono IP es un aparato telefónico con la misma apariencia física que los teléfonos tradicionales especialmente diseñado para conectarse a una red de telefonía IP.

<sup>5</sup> Los SoftPhone son programas para emular equipos dedicados (físicos) de VoIP.

el servidor B con la extensión 5002 (ver figura 74), se procede a realizar la prueba de llamadas en el sistema.



**Figura 74. Sofphone CounterPatch**

La llamada será realizada desde la extensión 5002 hacia la extensión 4002, marcando como si fuese un teléfono celular como se muestra en la figura 75.



**Figura 75. Llamando de servidor A a servidor B**

Dado este escenario se inicia el script esperar en el nodo que simula el teléfono IP (SofPhone) y se verifica que el puerto en el que escucha el servicio iniciado corresponda con el indicado en el proceso de registro. Damos un clic para contestar, y se logra con éxito del registro de llamada entrante, lográndose la comunicación en ambos terminales.



Figura 76. Aceptando llamada



Figura 77. Comunicación de terminales servidor A y servidor B

## CONCLUSIONES

El presente Proyecto Fin de Carrera, pasaría a ser una herramienta más al proyecto principal “la Telemedicina” .Como ya vimos en capítulos anteriores el proyecto está dividido en dos partes. Una primera parte es el diseño electrónico, esta parte se encarga de filtrar y amplificar la señal capturada por el estetoscopio. Este diseño que se ha implementado, no cumple con la expectativa deseada, sumándose ruido a la salida. Además de no contar con un programa de visualización de las señales acústicas.

La segunda parte, es el proceso de la transmisión del sonido respiratorio captado por el estetoscopio, a través de la red de internet utilizando los códec de audio mediante la implementación de hasta tres centrales de Asterisk. Esta parte cumple con el objetivo, logrando con éxito la transmisión remota de la señal acústica respiratoria. A pesar que es un software complejo, su configuración no es de las más simples, pero luego de tratar de entender lo más básico, en el proyecto se ha hecho de una manera lo más sencilla e intuitiva posible, para futuros desarrolladores, logrando resultados satisfactorios.

Es impresionante la cantidad de funciones que Asterisk proporciona, al investigar sobre el tema, se puede observar que tiene todo lo necesario para ser una PBX (o muchas otras cosas) de alto rendimiento.

Asterisk crece de manera rápida, y cada vez más se van agregando funciones a través del CVS, los distintos desarrolladores a través del mundo agregan nuevo código o parches cada día.

Otra excelente característica es la gran cantidad de protocolos que soporta, y la variedad de usos que se le puede dar, con esto se puede amoldar a cualquier situación real.

Todo este conjunto de cualidades, ya sean técnicas, de documentación o portabilidad, a mi gusto hacen de Asterisk una aplicación de excelente calidad, y una alternativa real para soluciones de alto nivel aplicado a la telemedicina.

## RECOMENDACIONES

El proceso investigativo del “diseño de un sistema de adquisición y transmisión de sonidos respiratorios utilizando telefonía IP”, para evaluar patologías respiratorias en la región queda abierto a la continuidad y profundización del tema, el cual forma la infraestructura básica, en la adquisición de sonidos respiratorios de un proyecto ambicioso. Se puede diseñar un filtrado más complejo compuesto por varias fases de filtrado, incluso se podría estudiar el utilizar un filtro CIC para realizar un pre filtrado y reducir ruidos y captar las señales deseadas. Y para el seguimiento del paciente realizar una base de datos, de alto nivel y muy dinámico como es Node.js y poder realizar un procesamiento digital a las señales adquiridas.

Aunque la principal aportación fue la implementación de las centrales de telefonía utilizando Asterisk, nuestra universidad y autoridades pertinentes, deberían sacarle el máximo provecho, por contar esto con licencia GPL y los múltiples servicios con que cuenta, como herramienta a la telemedicina para el área de la salud, no solo buscando un interés económico sino también un objetivo social. Este ideal o utopía como lo llamarían algunos, permite desarrollar nuevos elementos para la ingeniería electrónica que todavía en nuestra región no se han desarrollado al máximo.

## BIBLIOGRAFÍA Y REFERENCIAS

1. **ARGENTE , Horacio A. y otros:** 2010. Semiología Médica Fisiopatología y Propedéutica, Tomo 1, Editorial Panamericana, Argentina.
2. **ARIAS, Jesús:** 2009, Electrónica Analógica (II). 2da. ed. The Population Council, México.
3. **COUGHLIN, Robert:** 1999. Amp. Operacionales; 5da. ed. Interamericana, México.
4. **COUGHLIN, R. y DRISCOLL, F.:** 1993. Amplificadores Operacionales y Circuitos Integrados Lineales, 1ra. ed. PRENTICE/HALL, México.
5. **MALVINO, Albert P.:** 2000. Principios de Electrónica. 6ta. Ed. McGRAW- HILL Interamericana de España, S.A.U.
6. **LLANIO, R. y PERDOMO G.:** 2003. PROPEDEÚTICA CLÍNICA Y SEMIOLOGÍA MÉDICA. Tomo 1. 2da. ed. Centro Nacional de Información de Ciencias Médicas, Ciudad de La Habana, Cuba.
7. **OLIVER, Alvaro:** 2012. Guía de Instalación y Configuración, Publicación Científica N° 001.
8. **RUBIO, Antonio M.:** 2012. Tarificador CEL, Publicación Científica Máster Propio en Administración, Comunicaciones y Seguridad Informática, Universidad de Almería.
9. **PALACIOS, Ignacio:** 2012. Codificación de Audio para Sonidos Cardiacos y Respiratorios Adquiridos con Estetoscopio Digital, proyecto fin de carrera Universidad Autónoma de Madrid.
10. **SUROS, Antonio y otros:** 2001. Semiología Médica y Técnica Exploratoria, 8va. Ed. Editorial Masson, España.
11. **VILLAR, F , JAREÑO, E y ÁLVAREZ, R:** 2010. Patología Respiratoria, Manual de procedimientos de diagnóstico y control. 2da. ed. Gráficas Enar, S.A. Neumo Madrid.
12. <http://www.asterisk.org/> y otros (último acceso diciembre del 2013).

## GLOSARIO DE TÉRMINOS MÉDICOS

- Estetoscopio: m. Med. Aparato destinado a auscultar los sonidos del pecho y otras partes del cuerpo, ampliándolos con la menor deformación posible.
- Auscultación: f. Med. Acción y efecto de auscultar.
- Neumonía: f. Med. Inflamación del pulmón o de una parte de él , parénquima pulmonar producida generalmente por el neumococo
- Bronquiolitis: infección que afecta a las vías respiratorias diminutas, denominadas "bronquiolos"
- Paramédico: Perteneciente o relativo al personal auxiliar en tareas médicas. U. t. c. s.
- Patología respiratoria: med. Parte de la medicina que estudia las enfermedades del sistema respiratorio.
- Tronco encefálico: es la unión del mesencéfalo, la protuberancia anular, el bulbo raquídeo. Es la mayor ruta de comunicación entre el cerebro anterior, la médula espinal y los nervios periféricos
- Metabolismo: Biol. Conjunto de reacciones químicas que efectúan constantemente las células de los seres vivos con el fin de sintetizar sustancias complejas a partir de otras más simples, o degradar aquellas para obtener estas.
- Sistema respiratorio: sistema que tiene la función de hacer llegar el oxígeno del aire al interior del organismo para que pueda ser transportado a todas las células del cuerpo mediante el sistema sanguíneo. El sistema respiratorio también se encarga de sacar fuera del cuerpo el dióxido de carbono o CO<sub>2</sub>, que es un gas de desecho del metabolismo celular
- Alveolos: Anat. Cada una de las fositas hemisféricas en que terminan las últimas ramificaciones de los bronquiolos.
- Diafragma: Anat. En el cuerpo de los mamíferos, membrana formada en su mayor parte por fibras musculares, que separa la cavidad torácica de la abdominal.

- Diafragma-estetoscopio: m. Separación, generalmente movable, que intercepta la comunicación entre dos partes de un aparato o de una máquina.
- Campana-estetoscopio: parte del estetoscopio que se usa para escuchar sonidos de baja frecuencia.
- Hematíes: Biol. Glóbulo rojo de la sangre. U. m. en pl.
- Capilares: Anat. Cada uno de los vasos muy finos que enlazan en el organismo las circulaciones arterial y venosa, formando redes.
- Egofonía: f. Med. Resonancia de la voz que se percibe al auscultar el tórax de los enfermos con derrame de la pleura.
- Broncofonía: f. aumento de la resonancia normal del pulmón mientras se hace hablar al enfermo durante su auscultación.



## ANEXOS





# INA128 INA129

SBOS051B - OCTOBER 1995 - REVISED FEBRUARY 2005

## Precision, Low Power INSTRUMENTATION AMPLIFIERS

### FEATURES

- LOW OFFSET VOLTAGE: 50µV max
- LOW DRIFT: 0.5µV/°C max
- LOW INPUT BIAS CURRENT: 5nA max
- HIGH CMR: 120dB min
- INPUTS PROTECTED TO ±40V
- WIDE SUPPLY RANGE: ±2.25V to ±18V
- LOW QUIESCENT CURRENT: 700µA
- 8-PIN PLASTIC DIP, SO-8

### APPLICATIONS

- BRIDGE AMPLIFIER
- THERMOCOUPLE AMPLIFIER
- RTD SENSOR AMPLIFIER
- MEDICAL INSTRUMENTATION
- DATA ACQUISITION

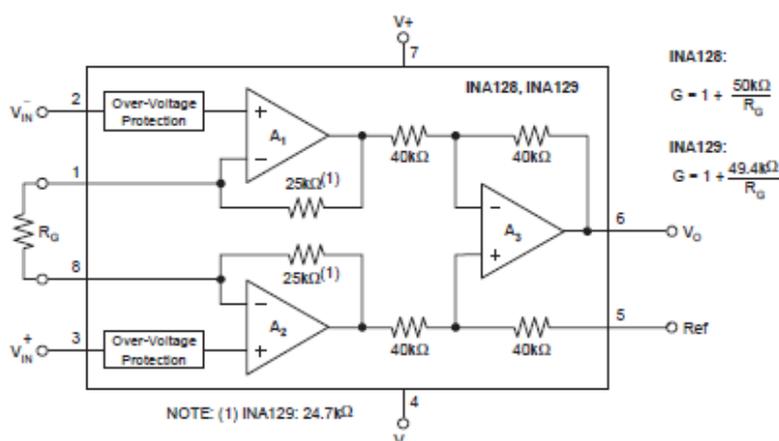
### DESCRIPTION

The INA128 and INA129 are low power, general purpose instrumentation amplifiers offering excellent accuracy. The versatile 3-op amp design and small size make them ideal for a wide range of applications. Current-feedback input circuitry provides wide bandwidth even at high gain (200kHz at G = 100).

A single external resistor sets any gain from 1 to 10,000. The INA128 provides an industry-standard gain equation; the INA129 gain equation is compatible with the AD620.

The INA128/INA129 is laser trimmed for very low offset voltage (50µV), drift (0.5µV/°C) and high common-mode rejection (120dB at G ≥ 100). It operates with power supplies as low as ±2.25V, and quiescent current is only 700µA—ideal for battery-operated systems. Internal input protection can withstand up to ±40V without damage.

The INA128/INA129 is available in 8-pin plastic DIP and SO-8 surface-mount packages, specified for the -40°C to +85°C temperature range. The INA128 is also available in a dual configuration, the INA2128.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

All trademarks are the property of their respective owners.

PRODUCTION DATA Information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



Copyright © 1995–2005, Texas Instruments Incorporated

**INA128  
INA129**



SBOS051B - OCTOBER 1995 - REVISED FEBRUARY 2005

**ABSOLUTE MAXIMUM RATINGS<sup>(1)</sup>**

Supply Voltage	±18V
Analog Input Voltage Range	±40V
Output Short-Circuit (to ground)	Continuous
Operating Temperature	-40°C to +125°C
Storage Temperature Range	-55°C to +125°C
Junction Temperature	+150°C
Lead Temperature (soldering, 10s)	+300°C

<sup>(1)</sup> Stresses above these ratings may cause permanent damage. Exposure to absolute maximum conditions for extended periods may degrade device reliability. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those specified is not implied.

**ELECTROSTATIC DISCHARGE SENSITIVITY**



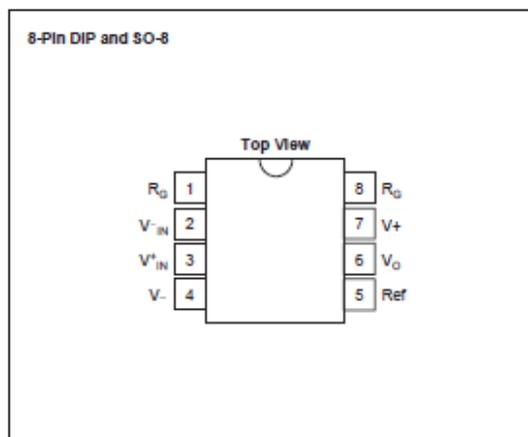
This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

**ORDERING INFORMATION**

For the most current package and ordering information, see the Package Option Addendum located at the end of this data sheet.

**PIN CONFIGURATION**





INA128  
INA129

GB09051B - OCTOBER 1995 - REVISED FEBRUARY 2005

**ELECTRICAL CHARACTERISTICS**

At  $T_A = +25^\circ\text{C}$ ,  $V_O = \pm 15\text{V}$ ,  $R_L = 10\text{k}\Omega$ , unless otherwise noted.

PARAMETER	CONDITIONS	INA128P, U INA129P, U			INA128PA, UA INA129PA, UA			UNIT	
		MIN	TYP	MAX	MIN	TYP	MAX		
<b>INPUT</b>									
Offset Voltage, RTI	$T_A = +25^\circ\text{C}$ $T_A = T_{\text{MIN}}$ to $T_{\text{MAX}}$ $V_B = \pm 2.25\text{V}$ to $\pm 18\text{V}$								
Initial			$\pm 10 \pm 100/\text{G}$	$\pm 50 \pm 500/\text{G}$		$\pm 25 \pm 100/\text{G}$	$\pm 125 \pm 1000/\text{G}$	$\mu\text{V}$	
vs Temperature				$\pm 0.2 \pm 2/\text{G}$	$\pm 0.5 \pm 20/\text{G}$		$\pm 0.2 \pm 5/\text{G}$	$\pm 1 \pm 20/\text{G}$	$\mu\text{V}/^\circ\text{C}$
vs Power Supply				$\pm 0.2 \pm 20/\text{G}$	$\pm 1 \pm 100/\text{G}$		*	$\pm 2 \pm 200/\text{G}$	$\mu\text{V}/\text{V}$
Long-Term Stability				$\pm 0.1 \pm 3/\text{G}$			*		$\mu\text{V}/\text{mo}$
Impedance, Differential			$10^{10} \parallel 2$			*		$\Omega \parallel \text{pF}$	
Common-Mode			$10^{11} \parallel 9$			*		$\Omega \parallel \text{pF}$	
Common-Mode Voltage Range <sup>(1)</sup>	$V_O = 0\text{V}$	(V+) - 2 (V-) + 2	(V+) - 1.4 (V-) + 1.7		*	*		V	
Safe Input Voltage				$\pm 40$			*	V	
Common-Mode Rejection	$V_{\text{CM}} = \pm 13\text{V}$ , $\Delta R_G = 1\text{k}\Omega$								
	G = 1	80	86		73	*		dB	
	G = 10	100	106		93	*		dB	
	G = 100	120	125		110	*		dB	
	G = 1000	120	130		110	*		dB	
<b>BIAS CURRENT</b>									
vs Temperature			$\pm 2$	$\pm 5$		*	$\pm 10$	nA	
Offset Current			$\pm 1$	$\pm 5$		*	$\pm 10$	pA/°C	
vs Temperature			$\pm 30$			*		nA	
			$\pm 30$			*		pA/°C	
<b>NOISE VOLTAGE, RTI</b>									
f = 10Hz	G = 1000, $R_G = 0\Omega$		10			*		$\text{nV}/\sqrt{\text{Hz}}$	
f = 100Hz			8			*		$\text{nV}/\sqrt{\text{Hz}}$	
f = 1kHz			8			*		$\text{nV}/\sqrt{\text{Hz}}$	
$f_B = 0.1\text{Hz}$ to 10Hz			0.2			*		$\mu\text{V}_{\text{pp}}$	
<b>Noise Current</b>									
f = 10Hz			0.9			*		$\text{pA}/\sqrt{\text{Hz}}$	
f = 1kHz			0.3			*		$\text{pA}/\sqrt{\text{Hz}}$	
$f_B = 0.1\text{Hz}$ to 10Hz			30			*		$\text{pA}_{\text{pp}}$	
<b>GAIN</b>									
Gain Equation, INA128			$1 + (50\text{k}\Omega/R_G)$			*		V/V	
INA129			$1 + (49.4\text{k}\Omega/R_G)$			*		V/V	
Range of Gain		1		10000	*		*	V/V	
Gain Error	G = 1		$\pm 0.01$	$\pm 0.024$		*	$\pm 0.1$	%	
	G = 10		$\pm 0.02$	$\pm 0.4$		*	$\pm 0.5$	%	
	G = 100		$\pm 0.05$	$\pm 0.5$		*	$\pm 0.7$	%	
	G = 1000		$\pm 0.5$	$\pm 1$		*	$\pm 2$	%	
Gain vs Temperature <sup>(2)</sup>	G = 1		$\pm 1$	$\pm 10$		*	*	ppm/°C	
50k $\Omega$ (or 49.4k $\Omega$ ) Resistance <sup>(2)(3)</sup>			$\pm 25$	$\pm 100$		*	*	ppm/°C	
Nonlinearity	$V_O = \pm 13.6\text{V}$ , G = 1		$\pm 0.0001$	$\pm 0.001$		*	$\pm 0.002$	% of FSR	
	G = 10		$\pm 0.0003$	$\pm 0.002$		*	$\pm 0.004$	% of FSR	
	G = 100		$\pm 0.0005$	$\pm 0.002$		*	$\pm 0.004$	% of FSR	
	G = 1000		$\pm 0.001$	(4)		*	*	% of FSR	

NOTE: \* Specification is same as INA128P, U or INA129P, U.

(1) Input common-mode range varies with output voltage — see typical curves.

(2) Specified by wafer test.

(3) Temperature coefficient of the 50k $\Omega$  (or 49.4k $\Omega$ ) term in the gain equation.

(4) Nonlinearity measurements in G = 1000 are dominated by noise. Typical nonlinearity is  $\pm 0.001\%$ .

**INA128  
INA129**



SBOS051B - OCTOBER 1995 - REVISED FEBRUARY 2005

**ELECTRICAL CHARACTERISTICS (continued)**

At  $T_A = +25^\circ\text{C}$ ,  $V_S = \pm 15\text{V}$ ,  $R_L = 10\text{k}\Omega$ , unless otherwise noted.

PARAMETER	CONDITIONS	INA128P, U INA129P, U			INA128PA, UA INA129PA, UA			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
<b>OUTPUT</b>								
Voltage: Positive	$R_L = 10\text{k}\Omega$	(V+) - 1.4	(V+) - 0.9		*	*		V
Negative	$R_L = 10\text{k}\Omega$	(V-) + 1.4	(V-) + 0.8		*	*		V
Load Capacitance Stability			1000		*	*		pF
Short-Circuit Current			+6/-15		*	*		mA
<b>FREQUENCY RESPONSE</b>								
Bandwidth, -3dB	G = 1		1.3		*	*		MHz
	G = 10		700		*	*		kHz
	G = 100		200		*	*		kHz
	G = 1000		20		*	*		kHz
Slew Rate	$V_O = \pm 10\text{V}$ , G = 10		4		*	*		V/ $\mu\text{s}$
Settling Time, 0.01%	G = 1		7		*	*		$\mu\text{s}$
	G = 10		7		*	*		$\mu\text{s}$
	G = 100		9		*	*		$\mu\text{s}$
	G = 1000		80		*	*		$\mu\text{s}$
Overload Recovery	50% Overdrive		4		*	*		$\mu\text{s}$
<b>POWER SUPPLY</b>								
Voltage Range		$\pm 2.25$	$\pm 15$	$\pm 18$	*	*	*	V
Current, Total	$V_{IN} = 0\text{V}$		$\pm 700$	$\pm 750$	*	*	*	$\mu\text{A}$
<b>TEMPERATURE RANGE</b>								
Specification		-40		+85	*	*	*	$^\circ\text{C}$
Operating		-40		+125	*	*	*	$^\circ\text{C}$
$R_{JA}$ 8-Pin DIP			80		*	*	*	$^\circ\text{C}/\text{W}$
SO-8 SOIC			150		*	*	*	$^\circ\text{C}/\text{W}$

NOTE: \* Specification is same as INA128P, U or INA129P, U.  
 (1) Input common-mode range varies with output voltage — see typical curves.  
 (2) Specified by wafer test.  
 (3) Temperature coefficient of the 50k $\Omega$  (or 49.4k $\Omega$ ) term in the gain equation.  
 (4) Nonlinearity measurements in G = 1000 are dominated by noise. Typical nonlinearity is  $\pm 0.001\%$ .





UA741

GENERAL PURPOSE SINGLE OPERATIONAL AMPLIFIER

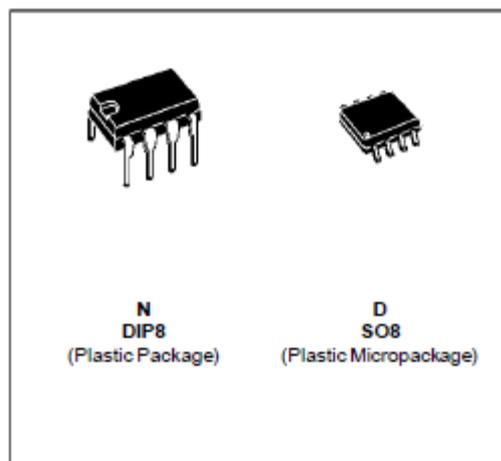
- LARGE INPUT VOLTAGE RANGE
- NO LATCH-UP
- HIGH GAIN
- SHORT-CIRCUIT PROTECTION
- NO FREQUENCY COMPENSATION REQUIRED
- SAME PIN CONFIGURATION AS THE UA709

DESCRIPTION

The UA741 is a high performance monolithic operational amplifier constructed on a single silicon chip. It is intended for a wide range of analog applications.

- Summing amplifier
- Voltage follower
- Integrator
- Active filter
- Function generator

The high gain and wide range of operating voltages provide superior performances in integrator, summing amplifier and general feedback applications. The internal compensation network (6dB/octave) insures stability in closed loop circuits.

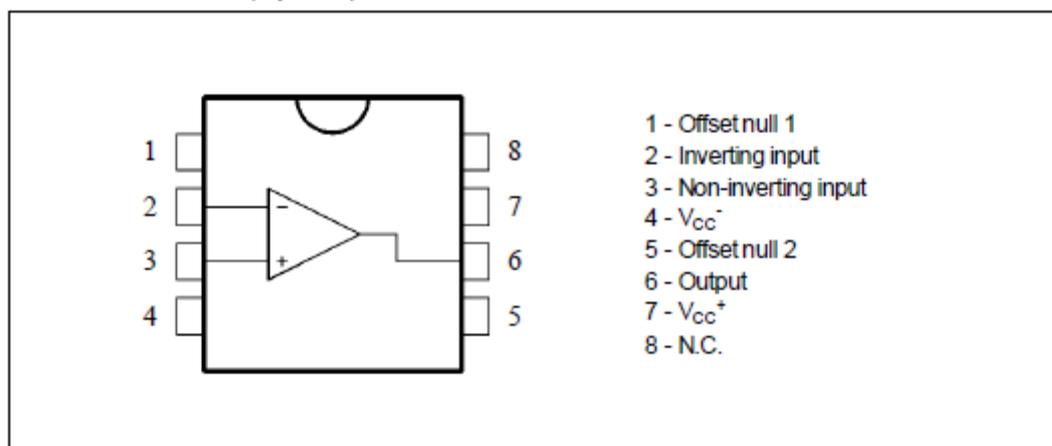


ORDER CODES

Part Number	Temperature Range	Package	
		N	D
UA741C	0°C, +70°C	•	•
UA741I	-40°C, +105°C	•	•
UA741M	-55°C, +125°C	•	•

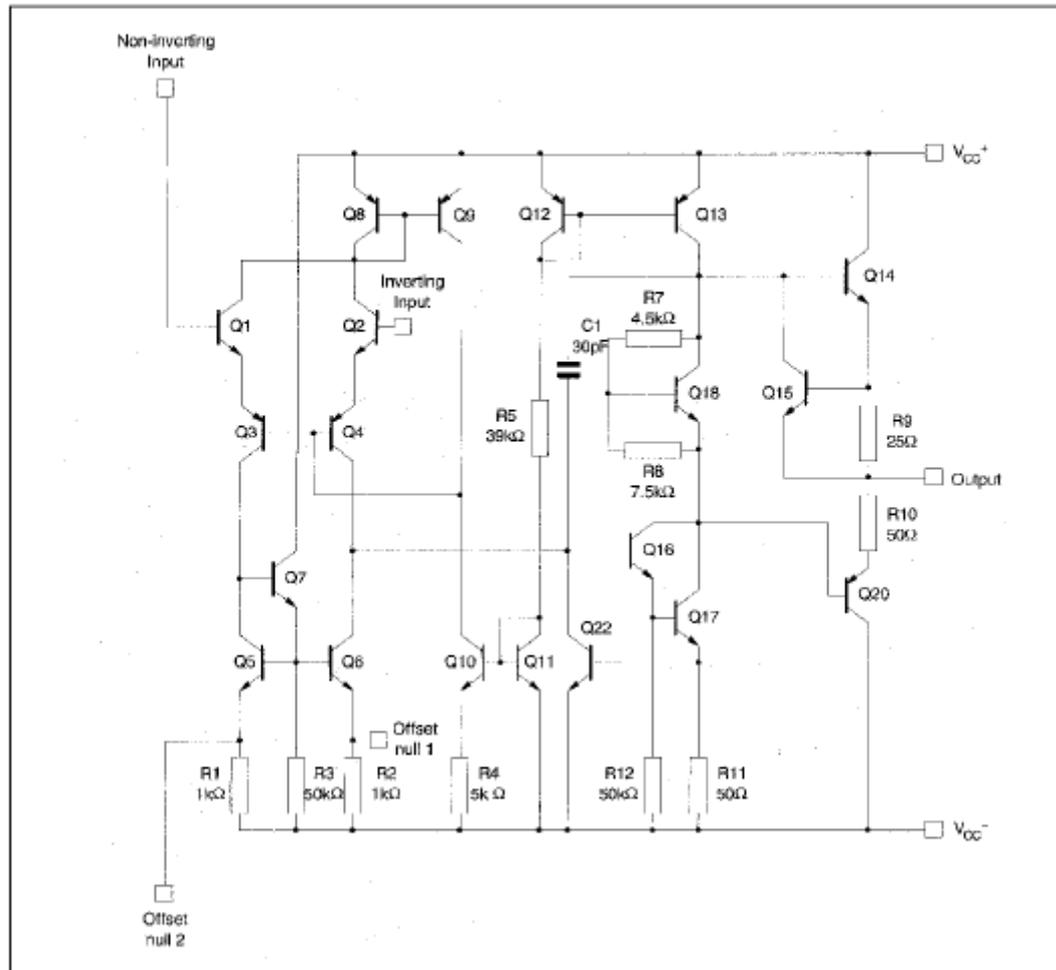
Example : UA741CN

PIN CONNECTIONS (top view)



UA741

**SCHEMATIC DIAGRAM**



**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	UA741M	UA741I	UA741C	Unit
$V_{cc}$	Supply Voltage		$\pm 22$		V
$V_{id}$	Differential Input Voltage		$\pm 30$		V
$V_i$	Input Voltage		$\pm 15$		V
$P_{tot}$	Power Dissipation		500		mW
	Output Short-circuit Duration		Infinite		
$T_{oper}$	Operating Free Air Temperature Range	-55 to +125	-40 to +105	0 to +70	$^{\circ}C$
$T_{stg}$	Storage Temperature Range	-65 to +150	-65 to +150	-65 to +150	$^{\circ}C$

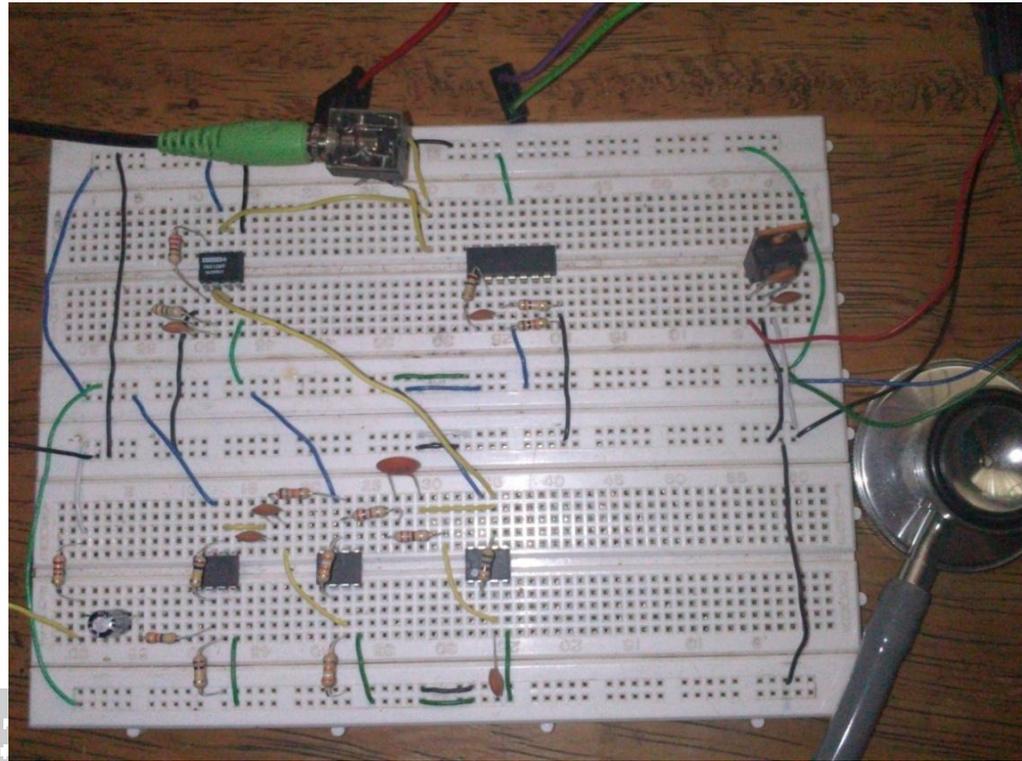
**ELECTRICAL CHARACTERISTICS**

$V_{CC} = \pm 15V$ ,  $T_{amb} = +25^{\circ}C$  (unless otherwise specified)

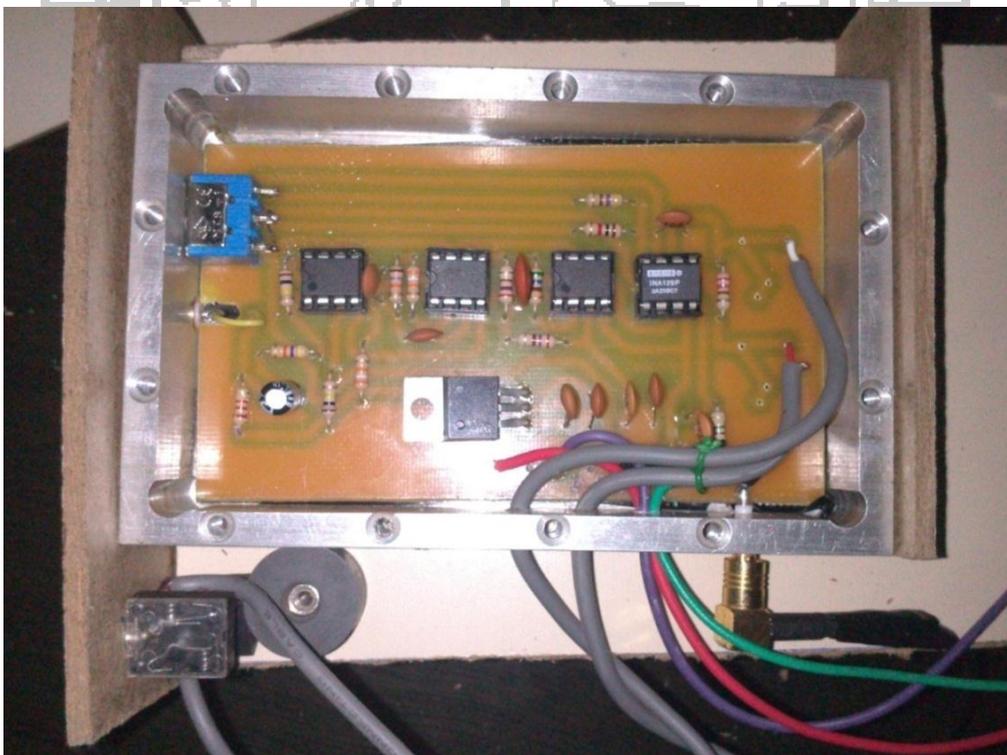
Symbol	Parameter	Min.	Typ.	Max.	Unit
$V_{io}$	Input Offset Voltage ( $R_S \leq 10k\Omega$ ) $T_{amb} = +25^{\circ}C$ $T_{min.} \leq T_{amb} \leq T_{max.}$		1	5 6	mV
$I_{io}$	Input Offset Current $T_{amb} = +25^{\circ}C$ $T_{min.} \leq T_{amb} \leq T_{max.}$		2	30 70	nA
$I_b$	Input Bias Current $T_{amb} = +25^{\circ}C$ $T_{min.} \leq T_{amb} \leq T_{max.}$		10	100 200	nA
$A_{vd}$	Large Signal Voltage Gain * ( $V_o \pm 10V$ , $R_L = 2k\Omega$ ) $T_{amb} = +25^{\circ}C$ $T_{min.} \leq T_{amb} \leq T_{max.}$	50 25	200		V/mV
SVR	Supply Voltage Rejection Ratio ( $R_S \leq 10k\Omega$ ) $T_{amb} = +25^{\circ}C$ $T_{min.} \leq T_{amb} \leq T_{max.}$	77 77	90		dB
$I_{cc}$	Supply Current, no load $T_{amb} = +25^{\circ}C$ $T_{min.} \leq T_{amb} \leq T_{max.}$		1.7	2.8 3.3	mA
$V_{icm}$	Input Common Mode Voltage Range $T_{amb} = +25^{\circ}C$ $T_{min.} \leq T_{amb} \leq T_{max.}$	$\pm 12$ $\pm 12$			V
CMR	Common-mode Rejection Ratio ( $R_S \leq 10k\Omega$ ) $T_{amb} = +25^{\circ}C$ $T_{min.} \leq T_{amb} \leq T_{max.}$	70 70	90		dB
$I_{os}$	Output Short-circuit Current	10	25	40	mA
$\pm V_{OPP}$	Output Voltage Swing $T_{amb} = +25^{\circ}C$ $T_{min.} \leq T_{amb} \leq T_{max.}$ $R_L = 10k\Omega$ $R_L = 2k\Omega$ $R_L = 10k\Omega$ $R_L = 2k\Omega$	12 10 12 10	14 13		V
SR	Slew Rate ( $V_i = \pm 10V$ , $R_L = 2k\Omega$ , $C_L = 100pF$ , $T_{amb} = 25^{\circ}C$ , unity gain)	0.25	0.5		V/ $\mu s$
$t_r$	Rise Time ( $V_i = \pm 20mV$ , $R_L = 2k\Omega$ , $C_L = 100pF$ , $T_{amb} = 25^{\circ}C$ , unity gain)		0.3		$\mu s$
$K_{OV}$	Overshoot ( $V_i = 20mV$ , $R_L = 2k\Omega$ , $C_L = 100pF$ , $T_{amb} = 25^{\circ}C$ , unity gain)		5		%
$R_i$	Input Resistance	0.3	2		M $\Omega$
GBP	Gain Bandwidth Product ( $V_i = 10mV$ , $R_L = 2k\Omega$ , $C_L = 100pF$ , $f = 100kHz$ )	0.7	1		MHz
THD	Total Harmonic Distortion ( $f = 1kHz$ , $A_v = 20dB$ , $R_L = 2k\Omega$ , $V_o = 2V_{PP}$ , $C_L = 100pF$ , $T_{amb} = 25^{\circ}C$ )		0.08		%
$e_n$	Equivalent Input Noise Voltage ( $f = 1kHz$ , $R_s = 100\Omega$ )		23		$\frac{nV}{\sqrt{Hz}}$
$\phi_m$	Phase Margin		50		Degrees

**ANEXO II: Imágenes del ensamble y funcionamiento del proyecto.**





**Figura 78. Prueba del circuito en protoboard**



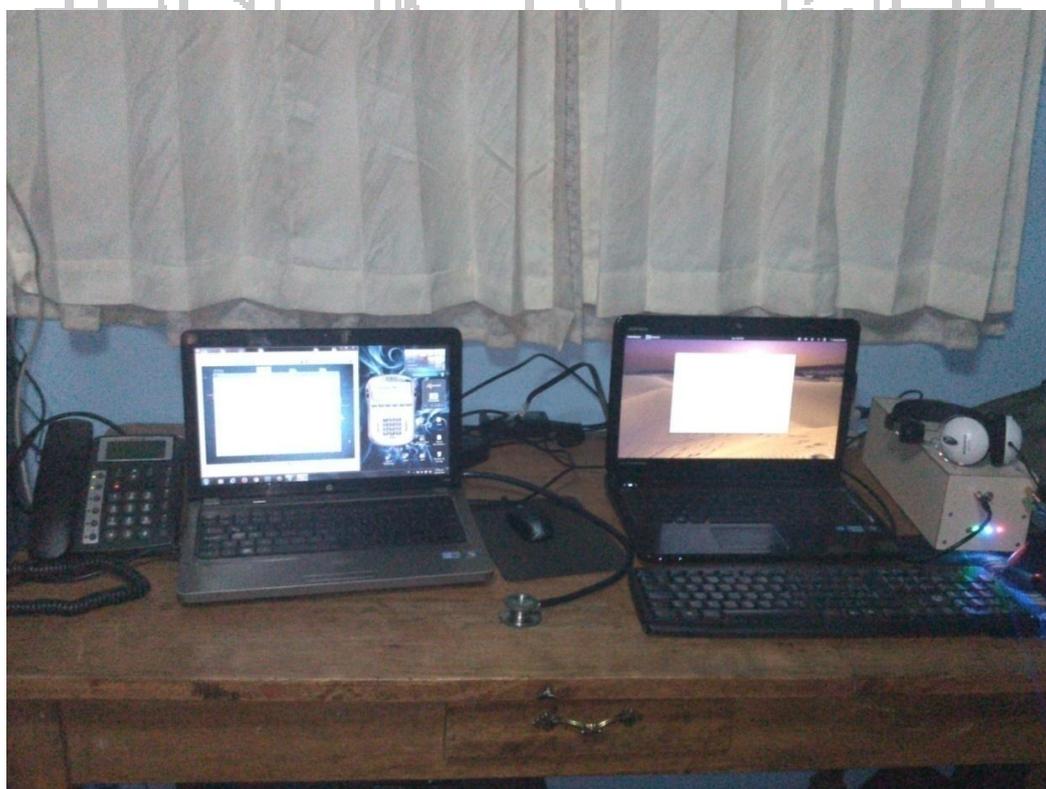
**Figura 79. Ensamble del circuito**

```

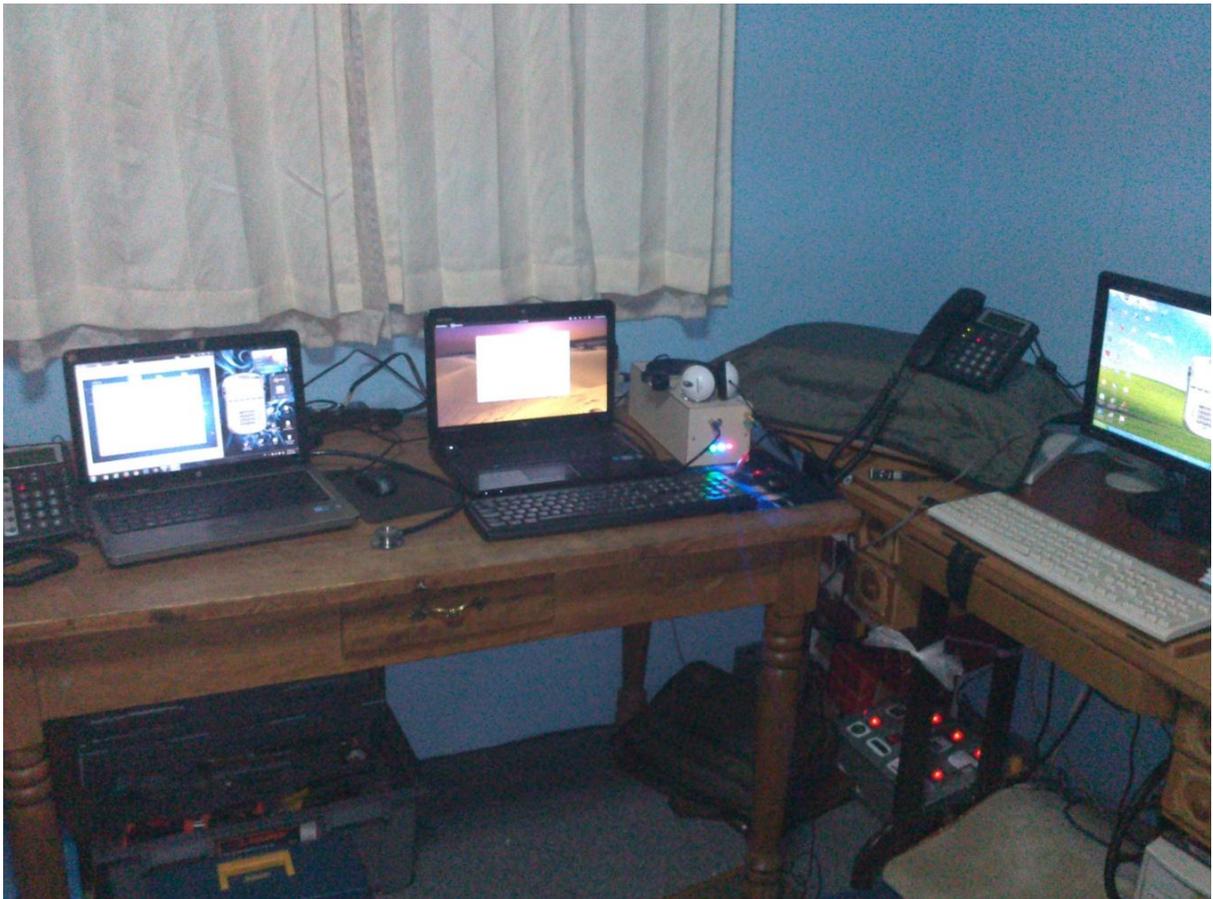
Asterisk Console on 'Bruno' (pid 23807)
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/asterisk/asterisk.conf
;nofork = yes ; Same as -f at startup.
;quiet = yes ; Same as -q at startup.
;timestamp = yes ; Same as -T at startup.
;execincludes = yes ; Support #exec in config files.
;console = yes ; Run as console (same as -c at startup).
;highpriority = yes ; Run realtime priority (same as -p at
; startup).
;initcrypto = yes ; Initialize crypto keys (same as -i at
; startup).
;nocolor = yes ; Disable console colors.
;dontwarn = yes ; Disable some warnings.
;dumpcore = yes ; Dump core on crash (same as -g at startup).
;languageprefix = yes ; Use the new sound prefix path syntax.
;internal_timing = yes
;systemname = my_system_name ; Prefix uniqueid with a system name for
; Global uniqueness issues.
;autosystemname = yes ; Automatically set systemname to hostname,
; uses 'localhost' on failure, or systemname if
; set.
;maxcalls = 10 ; Maximum amount of calls allowed.
;maxload = 0.9 ; Asterisk stops accepting new calls if the
; load average exceed this limit.
;maxfiles = 1000 ; Maximum amount of openfiles.
;minmemfree = 1 ; In MBs, Asterisk stops accepting new calls if
; the amount of free memory falls below this
; watermark.
;cache_record_files = yes ; Cache recorded sound files to another
; directory during recording.
;record_cache_dir = /tmp ; Specify cache directory (used in conjunction
; with cache_record_files)...
;transmit_silence = yes ; Transmit silence while a channel is in a
; waiting state, a recording only state, or

```

**Figura 80. Compilando Asterisk**



**Figura 81. Prueba de servidores Asterisk**



**Figura 82. Funcionamiento del proyecto**

