

UNIVERSIDAD TECNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA EN
SISTEMAS COMPUTACIONALES

TEMA:

Diseño e Implementación de un sistema de consultas de consumo de agua potable mediante TTSIVR, tecnologías web y mensajes de texto para el Gobierno Municipal de Antonio Ante utilizando herramientas open source.

AUTOR

Darwin Olayo Hernández Carrillo

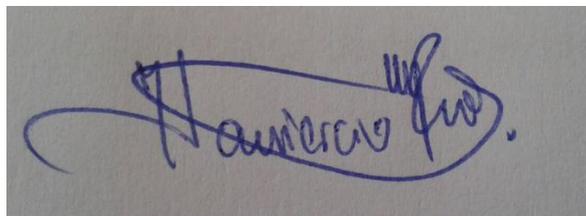
DIRECTOR

Ing. Mauricio Rea

Ibarra – Ecuador
2015

CERTIFICACIÓN

Certifico que la Tesis **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONSULTAS DE CONSUMO DE AGUA POTABLE MEDIANTE TTSIVR, TECNOLOGÍAS WEB Y MENSAJES DE TEXTO PARA EL GOBIERNO MUNICIPAL DE ANTONIO ANTE UTILIZANDO HERRAMIENTAS OPEN SOURCE”** ha sido realizada en su totalidad por la señor Hernández carrillo Darwin Olayo portador de la cédula de identidad número: 1002610069

A handwritten signature in blue ink, appearing to read "Mauricio Rea", is centered on a light gray background.

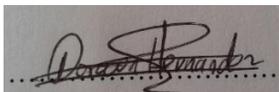
Ing. Mauricio Rea
DIRECTOR DE TESIS

UNIVERSIDAD TÉCNICA DEL NORTE**CESION DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

Yo, Hernández Carrillo Darwin Olayo con cedula de identidad Nro. 1002610069, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, articulo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONSULTAS DE CONSUMO DE AGUA POTABLE MEDIANTE TTSIVR, TECNOLOGÍAS WEB Y MENSAJES DE TEXTO PARA EL GOBIERNO MUNICIPAL DE ANTONIO ANTE UTILIZANDO HERRAMIENTAS OPEN SOURCE”**, que ha sido desarrollada para optar por el título de Ingeniería en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte



.....
Firma

Nombre: Hernández Carrillo Darwin Olayo

Cédula: 1002610069

Ibarra a los 23 días del mes de Julio del 2015

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente investigación:

DATOS DE CONTACTO	
CEDULA DE IDENTIDAD	1002610069
APELLIDOS Y NOMBRES	Hernandez Carrillo Darwin Olayo
DIRECCIÓN	Ibarra av. Atahualpa 27-70 y los Incas
EMAIL	darwinolayo@gmail.com
TELÉFONO FIJO	062651011
TELÉFONO MOVIL	0986545583
DATOS DE LA OBRA	
TITULO	“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONSULTAS DE CONSUMO DE AGUA POTABLE MEDIANTE TTSIVR, TECNOLOGÍAS WEB Y MENSAJES DE TEXTO PARA EL GOBIERNO MUNICIPAL DE ANTONIO ANTE UTILIZANDO HERRAMIENTAS OPEN SOURCE”
AUTOR	Hernandez Carrillo Darwin Olayo
FECHA	23 julio de 2015
PROGRAMA	PREGRADO
TITULO POR EL QUE	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR	Ing. Mauricio Rea

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Hernández Carrillo Darwin Olayo, con cedula de identidad Nro. 1002610069, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 143.



Nombre: Hernández Carrillo Darwin Olayo

Cédula: 1002610069

Ibarra a los 23 días del mes de Julio del 2015

DEDICATORIA

A mis padres y hermanos, ejemplos de superación y constancia, gracias por su infinito apoyo, colaboración y consejos durante todas las etapas de mi vida.

Darwin H.

AGRADECIMIENTO

A DIOS por su infinita bondad, por el regalo de la vida y la oportunidad de superación tras cada meta trazada y cumplida.

Al Gobierno Municipal de Antonio Ante por la colaboración prestada durante el desarrollo del presente proyecto en especial al Ing. Francisco Arteaga Director del Departamento de Sistemas.

A la Universidad Técnica del Norte, docentes, amigos y demás personas que de una u otra forma me han apoyado y colaborado con sus consejos y enseñanzas a lo largo de mi etapa universitaria.

Y un agradecimiento muy especial al Ing. Mauricio Rea por su guía y aporte incondicional en la culminación del presente Trabajo de Grado.

RESUMEN

El principal objetivo de este proyecto de tesis, es presentar una solución que pueda mejorar la atención a los usuarios del Gobierno Municipal de Antonio Ante y por ende de la empresa Pública de Agua Potable y Alcantarillado, atreves de diferentes medios de acceso como son la telefonía y la web, de esta manera disminuir la aglomeración de usuarios en las dependencias municipales, al momento de hacer consultas del consumo de planillas de agua potable.

Con la utilización de Asterisk vamos a guiar paso a paso la instalación ,configuración y puesta en marcha de una PBX, en la cual vamos a crear nuestro TTS IVR el cual es un valor agregado a la central telefonía IP. Además veremos de cómo lograr la conexión con la base de datos SQL SERVER para lograr la extracción de la información que necesitamos.

Así como también veremos la utilización de tecnologías web para el uso de consultas de las planillas de agua potable.

Una vez concluido el proyecto de tesis veremos los beneficios que brinda a todos los usuarios como también al personal administrativo.

ABSTRACT

The main objective of this thesis project is to present a solution that can improve service to users of the Municipal Government of Antonio Ante and therefore of the public enterprise Water and Sewage, dare different means of access as are the telephone and the web, thus decreasing the agglomeration of users at municipal offices, at the time of consultations consumption of drinking water rates.

By using Asterisk we will guide you step by step installation, configuration and implementation of a PBX, in which we create our TTS IVR which is an added value to the central IP telephony. Also we see how to achieve the connection to the database SQL SERVER to achieve extraction of the information we need.

And also we see the use of web technologies for use query payrolls water.

Once completed thesis project will see the benefits it provides to all users as well as administrative staff.

TABLA DE CON TENIDOS

Contenido

INTRODUCCION.....	15
1 CAPITULO I:.....	15
1.1 PROBLEMA.....	15
1.2 SITUACIÓN ACTUAL.....	16
1.3 OBJETIVOS	16
1.4 ALCANCE DEL PROYECTO.....	17
2 CAPITULO II.....	18
2.1 MARCO TEORICO	18
2.1.1 INTRODUCCION A LA TELEFONÍA	18
2.1.2 HISTORIA DE LA TELEFONÍA.....	18
2.1.3 PRINCIPIOS Y TRANSMISIÓN DE LA VOZ HUMANA	20
2.1.4 DIGITALIZACIÓN DELA VOZ.....	21
2.1.5 REDES ORIENTADAS A CIRCUITOS Y PAQUETES.....	22
2.1.6 RED DE TELEFONÍA PÚBLICA.....	23
2.1.7 RED DE TELEFONÍA CELULAR	24
2.2 INTRODUCCION A LA VOIP	25
2.2.1 PROTOCOLO IP.....	25
2.2.2 PROTOCOLOS DE TRANSPORTE.....	25
2.2.3 CODIFICACIÓN DE VOZ.....	26
2.3 INTRODUCCION A ASTERISK.....	26
2.3.1 QUE ES ASTERISK?	26

2.3.2	POR QUE ASTERISK?	27
2.3.3	ARQUITECTURA Y CARACTERISTICAS	28
2.3.4	HARDWARE PARA ASTERISK	28
2.4	CONVERSION DE TEXTO A VOZ, SISTEMA DE VOZ INTERACTIVA Y MENSAJES DE TEXTO.	29
2.4.1	QUE ES UN TTS, IVR Y SMS.....	29
2.4.2	SISTEMA DE RESPUESTA DE VOZ INTERACTIVA TRADICIONAL .	31
2.4.3	SISTEMA DE RESPUESTA DE VOZ INTERACTIVA DE ASTERISK ...	31
2.4.4	COMPONENTES DE UN TTS, IVR Y SMS.....	31
2.4.5	DISEÑO Y CONSIDERACIONES	32
2.4.6	MÓDULOS DE ASTERISK PARA CONSTRUIR TTS IVR.....	34
2.4.7	TTS IVR CON BASE DE DATOS	35
2.5	HERRAMIENTAS DE DESARROLLO WEB	39
2.5.1	DISEÑO Y MAQUETACIÓN.....	40
2.5.2	DESARROLLO WEB EN PHP y ASP.NET	41
3	IMPLEMENTACION	45
3.1	INSTALACIÓN DE LIBRERÍAS BASE PARA ASTERISK.....	46
3.2	INSTALACIÓN DE ASTERISK	67
3.3	INSTALACIÓN DE FREEPBX.....	68
3.4	INSTALACIÓN DE FESTIVAL	71
3.5	CONFIGURACIÓN DE FREEPBX COMO CENTRAL TELEFONICA	72
3.6	CONFIGURACIÓN DE ARCHIVOS DE ASTERISK	84
3.7	PROGRAMACIÓN DEL TTS IVR Y ENVIÓ DE MENSAJES DE TEXTO	91
3.8	ANÁLISIS DE REQUERIMIENTOS DE LA APLICACIÓN WEB DE CONSULTA DEL COMSUMO DE AGUA POTABLE.	98

3.9	DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN	99
4	CAPÍTULO: CONCLUSIONES Y RECOMENDACIONES	100
4.1	CONCLUSIONES	100
4.2	RECOMENDACIONES.....	101
5	GLOSARIO DE TÉRMINOS.....	102
	BIBLIOGRAFIA	104

LISTA DE FIGURAS

Figura 1: Alcance del Proyecto Fuente: Propia	18
Figura 2: Teléfono de Johann Philipp Reis Fuente: Wikipedia.....	19
Figura 3: Micrófono de carbón extraído de un teléfono de disco Fuente: Comunicaciones Unificadas.....	21
Figura 4: Simulación grafica de envío de paquetes en una red de paquetes Fuente: Asterisk PBX instalación configuración y puesta en marcha	22
Figura 5: Diagrama de red PSTN Fuente: TOMASI, Wayne. “Sistemas de comunicaciones electrónicas”, Cuarta edición. Ed.	24
Figura 6: Diagrama de red de telefonía celular Fuente: http://tecnocomunicaciones.wikispaces.com	24
Figura 7: Arquitectura de Asterisk Fuente: www.wikiasterisk.com	28
Figura 8: Tarjeta E1 Fuente: https://www.digium.com/products/telephony-cards/digital/single-span	29
Figura 9: Grafica de diseño y maquetación Fuente: http://josejfernandez.com/categoria/desarrollo-web	41
Figura 10: Inicio de instalación Centos Fuente: Propia.....	47
Figura 11: verificación de cd de instalación Fuente: Propia	48
Figura 12: Logo de inicio de instalación Centos Fuente: Propia.....	48
Figura 13: Selección de idioma de instalación de Centos Fuente: Propia.....	49
Figura 14: Selección de idioma del teclado Fuente: Propia	50
Figura 15: Dispositivos de almacenamiento para la instalación de Centos Fuente: Propia .	51
Figura 16: Nombre de nuestro host Fuente: Propia.....	52
Figura 17: Indicamos Nuestra Ubicación de nuestro País Fuente: Propia	53
Figura 18: Digitamos nuestra contraseña de súper usuario Fuente: Propia.....	54
Figura 19: Indica el tipo de instalación en nuestro disco Fuente: Propia	55
Figura 20: Proceso que inicio a instalar todos los paquetes del sistema Fuente: propia	56
Figura 21: La instalación de Centos ha finalizado Fuente : Propia	57
Figura 22: Primer inicio de Centos Fuente: Propia	57

Figura 23: Nos pide loguearse para ingresar al sistema Fuente: Propia.....	58
Figura 24: Instrucción de para configurar la tarjeta de red Fuente: Propia	58
Figura 25: Archivo original de configuración de tarjeta de red Fuente: Propia	59
Figura 26: Archivo editado de configuración de tarjeta de red Fuente: Propia.....	59
Figura 27: Ping de prueba de salida a internet Fuente: Propia	60
Figura 28: Archivo de configuración Selinux módulo de seguridad Fuente: Propia	62
Figura 29: Archivo de configuración tftp para aprovisionamiento Fuente: Propia.....	63
Figura 30: Activación de módulos para la instalación de asterisk Fuente: Propia.....	67
Figura 31: Asterisk se a instalado Satisfactoriamente Fuente: Propia.....	68
Figura 32: Pantalla principal de FreePBX Fuente: Propia	70
Figura 33: Estado del Sistema Fuente: Propia.....	73
Figura 34: Herramientas de administración Fuente: Propia	74
Figura 35: Aplicaciones para asterisk Fuente: Propia	74
Figura 36: Configuraciones Fuente: Propia.....	75
Figura 37: Anadir extensión Fuente: Propia.....	76
Figura 38: Creación de una extensión Fuente: Propia.....	77
Figura 39: Configuración tarjeta análoga Fuente: Propia.....	79
Figura 40: Configuración de tarjeta digium TDMA410 Fuente: Propia	80
Figura 41: Ingreso a la configuración de rutas entrantes Fuente: Propia	81
Figura 42: Llenamos los datos de nuestra ruta entrante Fuente: Propia.....	81
Figura 43: Pestaña para añadir ruta saliente Fuente: Propia.....	82
Figura 44: Archivo de configuración dahdi-channels.conf	83
Figura 45: Configuración de ruta Saliente Fuente: Propia	84
Figura 46: Archivo de configuración odbcinis.ini Fuente: Propia	85
Figura 47: Archivo de configuración freetds.conf Fuente: Propia	86
Figura 48: Archivo de configuración odbcinis.ini Fuente: Propia	87
Figura 49: Comando para comprobar la conexión con la base de datos MSSERVER Fuente Propia.....	87
Figura 50: Archivo de configuración extensions_custom.conf donde vamos a editar nuestro dial plan Fuente: Propia.....	88
Figura 51: Configuración de Custom Destinations para nuestro IVR Fuente: Propia	89

Figura 52: Descripción de cómo llegar a la pestaña de Misc Applications para nuestro IVR Fuente: Propia.....	89
Figura 53: Configuración de Misc Applications para nuestro IVR Fuente: Propia	90
Figura 54: Configuración de Misc Destinations Fuente: Propia	90
Figura 55: Diagrama de Flujo de IVR Fuente: Propia	91
Figura 56: Descripción de cómo llegar a la pestaña de IVR para configurar nuestro IVR Fuente: Propia.....	95
Figura 57: Pestaña de la opciones IVR para añadir nuestro primer IVR	96
Figura 58: Nos muestra todas las opciones para la configuración de nuestro IVR	96

INTRODUCCION

El continuo desarrollo de la tecnología en hardware y software, ha permitido que muchos servicios se junten como es el caso de las comunicaciones, entrega de información, etc. Esto ha generado una mejor administración centralizada y nos brinda mejores componentes de seguridad, también influye abaratando los costos de operación de las instituciones públicas o privadas.

Con este proyecto se propone una solución en conjunto o por separado de los diferentes aplicativos que nos permitirá lograr lo antes mencionado. Tal es el caso de Asterisk una solución Open Source de telefonía VoIP la cual provee de muchas características que solo existían en soluciones Privadas, como es TTS, IVR, video llamadas, buzón de llamadas, sala de conferencias, etc. Que se lo puede integrar con la red de datos reduciendo el coste de implementación y de mantenimiento. Además utilizaremos una solución de consultas web y de mensajes de texto, las cuales nos facilitarían mayor accesibilidad a la información de nuestros consumos de agua potable. Con lo cual ganaríamos tiempo y dinero tanto al usuario como a los funcionarios públicos.

1 CAPITULO I:

1.1 PROBLEMA

El Gobierno Municipal de Antonio Ante es una institución pública que está enfocada a brindar servicios a la comunidad para mejorar el modus vivendi de sus habitantes, de la misma manera se encuentra directamente inmerso en realizar las recaudaciones económicas e información de los servicios que proveen como: agua potable, catastros, patentes, etc.

Actualmente el Gobierno Municipal Antonio Ante genera la información y consultas sobre los pagos y servicios que brinda a la comunidad de una manera

localizada ya que los usuarios tienen que dirigirse a las ventanillas de cobro a realizar dichas consultas, generando una aglomeración de las personas en las filas.

Con estos antecedentes de la institución sobre el estado actual de su sistema de información se propone el diseño e implementación de un TTS (Text to speech) IVR basado en VoIP, tecnologías Web y mensajes de texto utilizando herramientas Open Source el cual automatizará el proceso de consultas de la información vía telefónica y web lo que además permitirá a los usuarios realizar todo tipo de consultas sin necesidad de acercarse a las ventanillas de cobro.

1.2 SITUACIÓN ACTUAL

Actualmente el Gobierno Municipal de Antonio Ante cuenta con un sistema de recaudación de las planillas de agua potable, el cual funciona de manera eficiente pero el usuario tiene que acercarse a la institución tanto para cancelar dichas planillas como para pedir información de cuanto está adeudando. Generando una aglomeración de usuarios en las ventanillas de recaudación.

1.3 OBJETIVOS

- Diseñar e implementar un sistema de consultas de planillas de consumo de agua potable mediante una Central de Telefonía IP, un sistema web y mensajes de texto, que facilite el acceso de forma automática a los datos que el usuario requiera mediante un sistema TTS IVR, a través del portal web del Municipio y mensajes de texto.
- Configurar una Central IP basada en Open Source (Asterisk).
- Definir un menú de opciones que nos permitirá obtener parámetros de preselección.
- Interconectar la central mediante un Dial Plan a la Base de Datos del Gobierno Municipal de Antonio Ante.

- Entregar al usuario la información en línea, en forma audible y a través mensajes de texto.
- Implementar un programa web que permita a los usuarios una consulta de planillas consumos rápida y directa.
- Resolver la aglomeración de usuarios en las ventanillas de recaudación del Gobierno municipal de Antonio Ante.

1.4 ALCANCE DEL PROYECTO

- El alcance de este proyecto de titulación es el diseño e implementación de un TTS IVR, de un sistema web y mensajes de texto para la realizar consultas de planillas de agua potable.
- El TTS IVR se realizará en cualquier distribución de Linux como Centos, Ubuntu, debían, etc. Asterisk, php y la base de datos SQL Server 2000 ya que el Gobierno Municipal de Antonio Ante cuenta con esta base de datos
- El sistema será realizado para la plataforma Linux
- EL sistema estará anexo a la PBX que actualmente funciona en la institución
- El sistema tendrá un límite de consultas dependiendo las líneas telefónicas
Se realizará el diseño del TTS IVR, el sistema de consultas web y mensajes de texto tomando en cuenta la integración con la base datos existentes en el Municipio de Antonio Ante
- Debido al recordé del presupuesto asignado a la dirección de TIC'S del Gobierno autónomo descentralizado de Antonio Ante la implementación de la consulta de planillas de agua potable por medio de mensajes de texto no es viable por el momento.

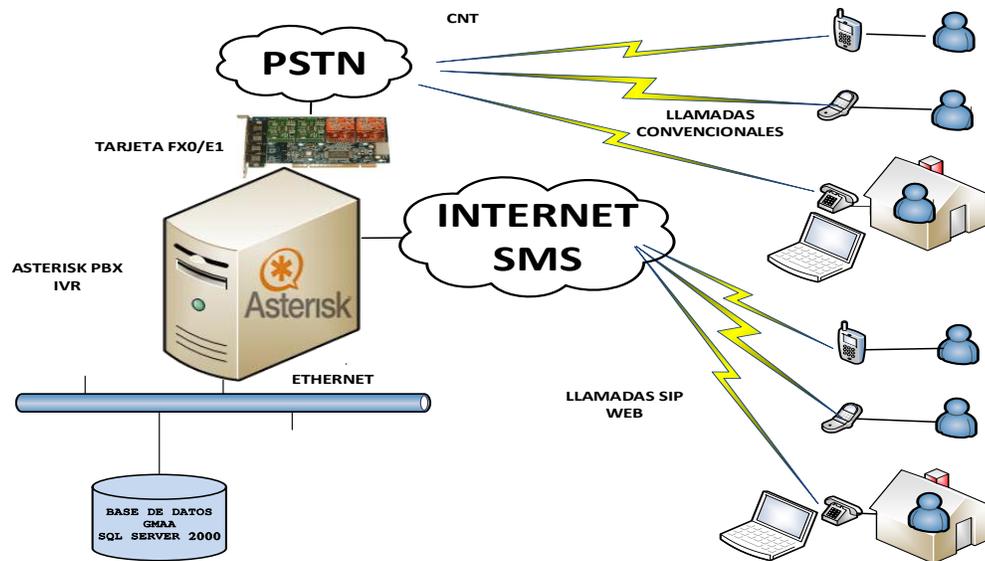


Figura 1: Alcance del Proyecto Fuente: Propia

2 CAPITULO II

2.1 MARCO TEORICO

2.1.1 INTRODUCCION A LA TELEFONÍA

En este capítulo se describirá una breve reseña de lo que es la telefonía, como funciona y sus diferentes tipos de redes telefónicas en la cuales se divide.

2.1.2 HISTORIA DE LA TELEFONÍA

Uno de los grandes inventos que perdura hasta la actualidad es la telefonía, cuyo origen data del siglo 19. Oír la voz del alguien remoto en tiempo real saliendo de una caja parecía algo mágico, pero fue a inicios de del siglo 19 que parecía algo

alcanzable, ya que se había descubierto la electricidad, el telégrafo y ya se experimentaba con ondas de radio.

Es 1849 Antonio Meucci médico italiano, llevo a cabo una demostración de un dispositivo capaz de transmitir la voz en la Habana. En 1954, el mismo en la ciudad de Nueva York hace una nueva demostración de su invento.

En 1860 el alemán Johann Philipp Reis construye un dispositivo capaz de transmitir la voz basada en la idea original de Charles Bourseul, quien en 1854 describió la construcción de dicho dispositivo pero nunca lo construyó. Johann continúa mejorando su dispositivo y un año más tarde estaba transmitiendo voz a más de 100 metros de distancia.

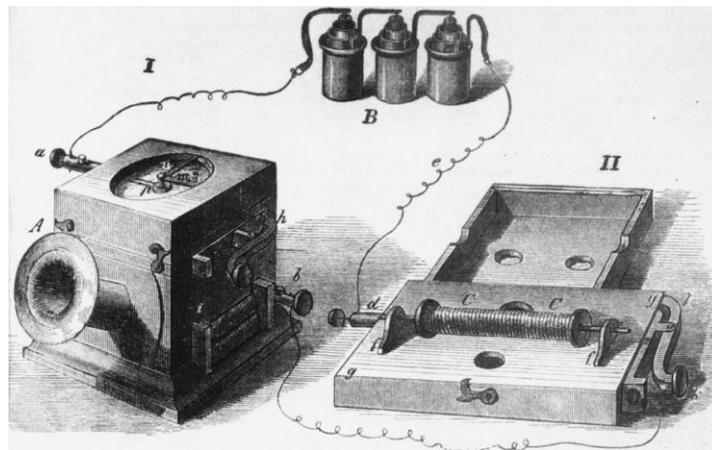


Figura 2: Teléfono de Johann Philipp Reis Fuente: Wikipedia

Más tarde Innocenzo Manzetti construye el telégrafo parlante el mismo que había visionado en 1844.

Hasta este momento existían varios prototipos pero nadie lo patentaba.

Meucci fue el primer en tratar de patentar su invento en 1871 pero por su condición económica no pudo terminar este trámite.

En 1875, un año después de expirar el trámite de Meucci, el escocés Alexander Graham Bell que radicaba en Estados Unidos, patenta un aparato similar.

En 1886, existían 150000 abonados telefónicos en Estados Unidos, a partir de entonces se empezó a convertir en un servicio básico de la sociedad actual. (Perez, 2014)

2.1.3 PRINCIPIOS Y TRANSMISIÓN DE LA VOZ HUMANA

La voz humana es un sonido emitido por una persona usando las cuerdas vocales. Para hablar cantar, reír, etc. Está compuesta por ondas acústicas que viajan por el aire a una velocidad de 1,244 km/h. puesto que la voz humana se atenúa rápidamente, pierde energía a medida que viaja por tanto luego de algunos metros no podemos escuchar una conversación. Antes que inventen el teléfono se conocían otros tipos de ondas como las electromagnéticas que podían ser transmitidas a través de un cable de cobre. Éste tipo de ondas es diferente a las ondas acústicas y viajaba la velocidad de la luz aproximadamente 300000km/s, casi instantáneamente desde un punto a otro. Además podemos controlar la atenuación de estas ondas y hacerlas recorrer grandes distancias. (Perez, 2014)

Tomando en cuenta estos hechos, es comprensible que muchos quieran transformar las ondas acústicas en ondas eléctricas y así poder transmitir la voz humana a través de conductores metálicos para poder cubrir grandes distancias.

Una característica importante es que las cuerdas vocales modulan la voz en un espectro que asila entre 20 Hz a 20Khz. Esto nos hace pensar que un micrófono debe ser capaz de capturar y transmitir todo este rango de frecuencias. En nuestros días sabemos que para transmitir voz entendible no se necesita transmitir todas las

frecuencias si no un rango de 400Hz a 4 KHz, esto distorsiona la voz ligeramente pero se puede entender. (Landívar, 2008)

Un elemento clave para invención del teléfono fue el micrófono que es el encargado de la conversión de ondas acústicas a ondas eléctricas



Figura 3: Micrófono de carbón extraído de un teléfono de disco Fuente: Comunicaciones Unificadas

2.1.4 DIGITALIZACIÓN DE LA VOZ

Digitalizar es tomar muestras de una señal de voz en intervalos de tiempo regulares de la amplitud de la señal analógica y transformar esta información a binario, este proceso se le conoce como muestreo, además de la etapa de muestreo interviene también interviene la cuantificación y codificación.

Cuantificación.- este proceso convierte las muestras analógicas en muestras que puedan tomar un conjunto discreto de valores. Estos valores se cuantifican en cantidades discretas, aun conjunto discreto de valores, se introduce naturalmente una distorsión a la señal original. Esta distorsión se conoce como ruido de cuantificación.

Codificación.- este proceso es asignar valores numéricos a los valores discretos.

El proceso de cuantificación y codificación adoptado en telefonía implementa un algoritmo no lineal, el cual dará una calidad de voz aceptable.

Harry Nyquist, ingeniero Suizo que trabajaba para AT&T, en 1928 resolvió cuanto es necesario muestra una señal como mínimo para poder reconstruirla luego de forma exacta ala original, el teorema decía que como mínimo se necesita el doble de ancho de banda como frecuencia de muestreo en la siguiente expresión $f_m \geq 2B_w$, por tanto deberíamos muestra a 8000Hz que es la frecuencia de muestreo que se usa en la mayoría de códec. (Landívar, 2008)

2.1.5 REDES ORIENTADAS A CIRCUITOS Y PAQUETES

Redes orientadas a circuitos son aquellas que establece un canal o circuito dedicado entre nodos antes de que se comuniquen, el resultado es el equivalente a conectar físicamente un par de cables de un extremo a otro, una vez conectado este canal o circuito no podrá ser usado por otros usuarios hasta que liberen el canal.

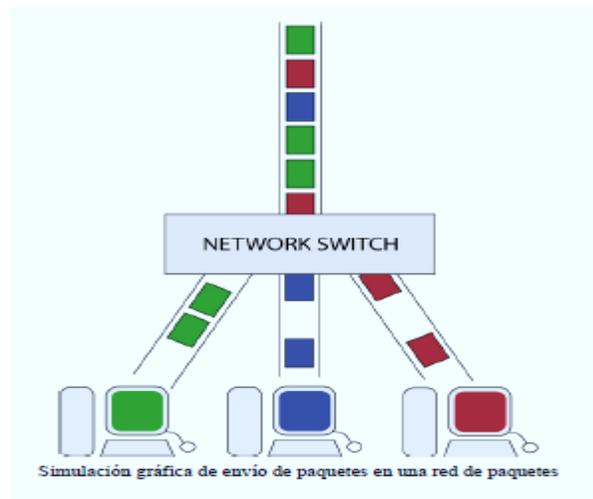


Figura 4: Simulación grafica de envío de paquetes en una red de paquetes Fuente: Asterisk PBX instalación configuración y puesta en marcha

Un ejemplo típico es la red del sistema telefónico tradicional la cual enlaza segmentos de cable para crear circuitos durante la duración de una llamada.

Redes orientadas a paquetes es el envío de datos en una red de computadoras, un paquete es un grupo de información que consta de los datos propiamente dicho y la información de control que indica la ruta a seguir hasta su destino, si el tamaño del paquete excede de un límite es necesario dividir en otros más pequeños y son re-ensamblados cuando llegan a su destino.

Un ejemplo son las redes IP es el caso del internet donde circulan diferentes tipos de información. (Perez, 2014)

2.1.6 RED DE TELEFONÍA PÚBLICA

También conocida como PSTN (Public Switched Telephone Network) por sus siglas en inglés, es una red de circuitos tradicional con conmutación optimizada para comunicaciones de voz en tiempo real. Es decir se trata de la telefonía que tenemos todos en nuestros hogares en la cual los teléfonos se conectan a una central de conmutación a través de un solo canal compartido por el micrófono y el auricular

Características

- Cada usuario se le asigna un circuito para señales analógicas con una banda base de 4KHz para cada conversación entre 2 usuarios.
- Red con cobertura nacional
- Es capaz de interconectarse con la red celular
- La tarifa del usuario depende de la distancia entre los extremos y la duración de la conexión

Está conformada de medios de transmisión y centrales de conmutación. Los medios de transmisión entre centrales se llaman troncales, actualmente transportan señales digitales por medios ópticos. (Landívar, 2008)

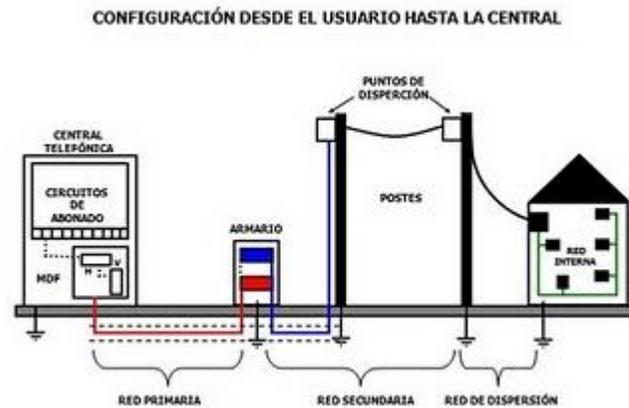


Figura 5: Diagrama de red PSTN Fuente: TOMASI, Wayne. "Sistemas de comunicaciones electrónicas", Cuarta edición. Ed.

2.1.7 RED DE TELEFONÍA CELULAR

También llamada red de telefonía móvil es la que no está conectada físicamente mediante cables si no que ocupa como medio de transmisión es el aire y los mensajes por medio de ondas electromagnéticas.

Para que esta red funcione necesita dos aspectos que son: la red de comunicaciones que está formada por antenas que las tenemos por todas las ciudades carreteras, etc. Y los terminales o celulares que poseemos todos, que ese medio por el cual accedemos a esta red, tanto las antenas y los celulares tiene dos funciones emisor-receptor y trabajan a una frecuencia entre 900 a 2000 MHz.

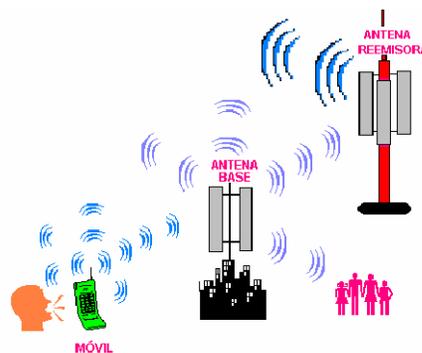


Figura 6: Diagrama de red de telefonía celular Fuente: <http://tecnocomunicaciones.wikispaces.com>

2.2 INTRODUCCION A LA VOIP

VOIP viene del inglés Voice Over Internet protocol, que en nuestro idioma significa voz sobre un protocolo de internet, básicamente la VOIP es un método el cual toma el audio análogo que escuchamos cuando hablamos por teléfono y lo transforma en datos digitales, en paquetes de datos, que van hacer transmitidos a través de internet o una red privada.

2.2.1 PROTOCOLO IP

El protocoló IP es la base fundamental de la internet, porta datagramas de la fuente al destino, su función principal es aceptar y transportar los datos provenientes de tcp o udp. Se caracteriza por:

- Protocolo orientado a no conexión y no confiable
- Fragmenta y re ensambla paquetes si es necesario
- Realiza funciones de enrutamiento

2.2.2 PROTOCOLOS DE TRANSPORTE

Este protocolo maneja el control de errores, flujo, la secuencia de paquetes, etc. Y existen tres protocolos.

TCP (transmission Control Protocol)

Este protocolo casi nunca usado para VOIP, ya que no garantiza que lleguen a su destino, TCP controla que los datos enviados no tengan errores y sean recibidos en el mismo orden que fueron enviados, si los datos no son recibidos o se pierden, son reenviados.

UDP (User Datagram Protocol)

A UDP no le importa si los datos llegan con errores o llegan en secuencia. UDP divide los datos en datagramas y son colocados en el medio para encontrar su destino final, sin ninguna confirmación si son recibidos o no.

SCTP (Stream Control Transmission Protocol)

SCTP fue creado para mejorar las deficiencias de los protocolos TCP y UDP, los objetivos son:

Mejorar las técnicas para evitar la congestión, estricta secuencia de datos enviados y una mínima latencia para mejorar la transmisión en tiempo real.

2.2.3 CODIFICACIÓN DE VOZ

Para transportar la voz se utilizan algunos protocolos como SIP, IAX y otros como RTP o RTCP. Como ya hemos visto la voz es una onda análoga y debe ser transformada a digital para ser enviada en una red IP mediante un formato adecuado llamados códec.

2.3 INTRODUCCION A ASTERISK

En este capítulo haremos una breve reseña de Asterisk su arquitectura y características

2.3.1 QUE ES ASTERISK?

Asterisk es un framework open source para aplicaciones de comunicaciones, convierte un computador común en un servidor de comunicaciones como son IP PBX, Gateway de VOIP, servidor de conferencias y otras soluciones personalizadas.

Asterisk se inició en 1999 cuando Mark Spencer lanzó su primera versión bajo licencia GPL. Desde entonces hasta la fecha se ha ido mejorando con la ayuda de Digium y la comunidad Asterisk.

2.3.2 POR QUE ASTERISK?

Por qué se a cuál sea la necesidad, Asterisk trabaja con líneas y terminales análogas, digitales de forma independiente o combinada. Además porque es compatible muchos protocolos de VoIP más usados como son:

SIP (protocolo de inicio de sesiones) fue creado para realizar sesiones interactivas entre dos o más usuarios, utilizando video, voz, mensajería instantánea, en la actualidad, son considerados como un estándar, utilizan el puerto 5060, su desventaja que no tiene un buen direccionamiento para el funcionamiento de NAT. (Russel Bryant, 2013).

IAX2 (inter Asterisk Exchange) es un protocolo libre creado por los fundadores de asterisk, utiliza el puerto UDP 4569 tanto para señalización de canal como para RTP (Protocolo de transporte en tiempo real), ofrece interoperabilidad con NAT / PAT / mascarada cortafuegos y otros servidores de seguridad, incluyendo la capacidad para realizar y recibir llamadas, y la transferencia de llamadas a otras estaciones. (asterisk.org, 2012)

H.323 en principio fue diseñada para el transporte de video conferencia, este protocolo está siendo remplazado por SIP ya está quedando obsoleto. Utiliza el puerto TCP 1720, es muy complejo su direccionamiento. (3cx.es, 2013)

MGCP es un protocolo de la capa de aplicación basado en texto para establecer y controlar las llamadas, está basado en la arquitectura maestro esclavo, y su origen está basado en cisco y bellcore.

SCCP (Skinny Call Control Protocol) es un protocolo propietario de Cisco es un conjunto de mensajes entre cliente y un call manager, un cliente skinny utiliza TCP/IP para conectarse a un callmanager, usa el puerto 2000. (Cisco, 2012)

También puede combinarse con cualquier servicio asociado que pueda imaginarse tanto como IVR con BDD, operadora virtual, videoconferencia, etc.

2.3.3 ARQUITECTURA Y CARACTERISTICAS

Asterisk está construido de manera modular, cada módulo tiene una función específica, ya sea a un controlador de canal o una conexión a una tecnología externa. Asterisk un variado y flexible set de características como funciones básicas como avanzadas de PBX, opera con sistemas estándar de telefonía y VOIP, Voice mail, conferencias, etc.

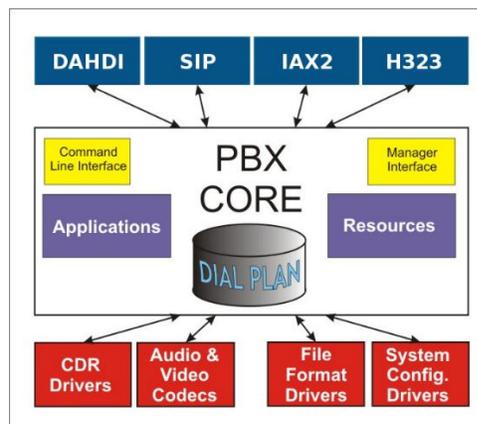


Figura 7: Arquitectura de Asterisk Fuente: www.wikiasterisk.com

2.3.4 HARDWARE PARA ASTERISK

Asterisk es capaz de comunicarse con muchas tecnologías diferentes hardware como son E1, Gateway, tarjetas FXS FXO, estas conexiones son a través de red

telefónica tradicional, como la PSNT necesitando hardware dedicado para su funcionamiento.

Muchas empresas fabrican este hardware, tales como Digium (patrocinador, propietario y desarrollador de Asterisk), Sangoma, Rhino, Open Vox y muchos más. Todas estas tienen diferentes requisitos de instalación y diferentes ubicaciones de archivos, un ejemplo de una tarjeta E1, que es un conjunto de 30 líneas digitales las cuales contratamos a nuestro operador telefónico.



Figura 8: Tarjeta E1 Fuente: <https://www.digium.com/products/telephony-cards/digital/single-span>

2.4 CONVERSION DE TEXTO A VOZ, SISTEMA DE VOZ INTERACTIVA Y MENSAJES DE TEXTO.

En este capítulo vamos a describir cada uno de los conceptos que intervienen para desarrollar nuestro proyecto.

2.4.1 QUE ES UN TTS, IVR Y SMS

CONVERSION DE TEXTO A VOZ

La voz es el principal medio de comunicación entre los seres humanos, y como ha ido evolucionando la tecnología, está a sido objeto de estudio y desarrollo de nuevas tecnologías, para mejorar la comunicación entre ordenadores y los humanos.

La conversión de texto a voz es un conjunto de procesamientos cuyo objetivo es permitir el control y el acceso a la información de ordenadores. Ya sea esta información datos de consumos como es el caso de agua, luz, teléfono. Etc.

El proceso de la conversión de texto a voz consiste que por medios automáticos de una voz artificial genera el sonido producido por una persona al leer un texto.

SISTEMA DE VOZ INTERACTIVA

En sigla conocida como IVR que consiste en la típica máquina que nos responde con mensajes pregrabados cuando llamamos a una empresa que posee una central telefónica, el IVR es un sistema telefónico el cual es capaz de recibir gran cantidad de llamadas e interactuar con los usuario a través de grabaciones de voz, al fin de reducir la necesidad de personal y los costos de servicios.

En un sistemas de respuesta interactiva está orientado a entregar información automatizada a través del teléfono permitiendo acceder a los servicios de información y operaciones autorizadas, las 24 horas del día desde cualquier parte del mundo.

Su funcionamiento creo que todos lo conocemos y si no es así lo vamos a describir:

Realizamos una llamada a un número de teléfono, el IVR contesta la llamada y le dice un menú de opciones por ejemplo “si quiere conocer su saldo marque 0”y siguiendo las instrucciones el usuario elige una opción digitando en el teléfono hasta conocer la información solicitada.

MENSAJES DE TEXTO

La gran mayoría de personas sabemos que es mensajes de texto (SMS o en ingles Short Message Service) son los que cuentan con características como su brevedad y fácil envío. Y no es más aquel mensaje que se recibe entre dispositivos móviles.

2.4.2 SISTEMA DE RESPUESTA DE VOZ INTERACTIVA TRADICIONAL

Los sistemas de respuesta interactiva o en siglas en ingles IVR (Interactive Voice Response), tradicionalmente utilizan hardware y software propietario, que dificultan la integración y personalización para realizar dichos sistemas telefónicos, lo cual representa costos elevados y se necesita ayuda de expertos.

En la actualidad existen empresas que se dedican a realizar sistemas de telefonía IVR muy robustos, que integran reconocimiento de voz, escritura y lectura de base de datos, conversión de texto a voz, etc., un ejemplo es la banca telefónica que casi todos alguna vez la hemos usado.

2.4.3 SISTEMA DE RESPUESTA DE VOZ INTERACTIVA DE ASTERISK

Como sabemos Asterisk nació por la necesidad de poder tener una PBX IP que podamos personalizar a nuestras necesidades, este software posee muchas funcionalidades que solo existían en PBX propietarias, tal es el caso de tener un sistema de respuesta de voz interactiva IVR.

En Asterisk tenemos la facilidad de crear IVR tan sencillos como es el caso de una contestadora automática la cual hemos escuchado la mayoría de personas cuando marcamos alguna empresa o institución pública y lo recibe con un saludo y le da opciones de números para marcar.

O crear verdaderos sistemas de respuesta interactiva con conexión a cualquier base de datos con la ayuda de scripts conocidos como AGI, que puede utilizar cualquier lenguaje de programación ya sea php, C, java, etc.

2.4.4 COMPONENTES DE UN TTS, IVR Y SMS

Conversión de texto a voz está compuesta por dos partes un front-end y un back-end. Front-end toma como entrada el texto que lo va a leer de un resultado hecha

por una consulta en nuestra base de datos. El back-end es el resultado en palabras que las personas podamos entender.

Sistema de voz interactivo (IVR) está compuesto menús los cuales interactúan con los usuarios mediante pulsaciones DTMF (multifrecuencia de doble tono) es un tipo de señalización que se utiliza en sistemas de telefonía de voz. Para saber que tecla fue presionada.

2.4.5 DISEÑO Y CONSIDERACIONES

Para nuestro diseño que será para hacer consultas de consumo de agua potable para el Municipio Autónomo Descentralizado de Antonio Ante hemos tomado las siguientes recomendaciones.

Objetivo. ¿Cuál va a ser el objetivo del IVR?, en nuestro caso el objetivo es dar a conocer la información del valor a pagar de su consumo de agua potable.

Mercado. ¿A qué mercado estará dirigido el IVR?, a toda la comunidad antena que cuenta con el servicio de agua potable

Orden. ¿En qué orden solicitan los clientes la información? Para nuestro IVR la información solicitada será el valor a cancelar por el consumo de agua potable y la fecha de vencimiento.

Además de estos aspectos debemos tomar en cuenta lo siguiente:

Principios generales:

- Proporcionar de la forma más simple la información que se solicita.
- Avisar al usuario si existe alguna restricción al principio del mensaje.
- Nuestro sistema debe interesar a todos los usuarios no solo a un cierto grupo.

- Hacer que nuestro sistema trabaje por el usuario, evitando ingresar el mismo dato varias veces.

Saludo inicial:

- En mensaje de bienvenida debe ser breve y claro: bienvenidos a....
- Es preciso evitar mensajes promocionales

Menú principal:

- El primer menú debe ser el principal a no ser que se tenga que pedirse identificación.
- Máximo 20 segundos de duración contando el saludo inicial y el menú principal.

Identificación:

- Pedir identificación sólo cuando sea necesaria.
- Pedir el identificador más fácil de recordar: cedula, teléfono,...
- Aceptar que el usuario dé cadenas variables de dígitos (ex.: la cedula puede tener 10 dígitos) sin causarle un error.
- Gestionar el final de la entrada de datos por medio de timeout, y solicitar una tecla adicional.

Menús:

- Debe estar provistos de una entradilla: título del menú que indica al usuario donde ha ido a parar y de qué tratan las opciones que escuchará.

Preguntas**Sí/No:**

Se usan para recuperar errores y para confirmar acciones por ejemplo si es correcto digite 1 caso contrario digite 2.

Errores de sustitución y falsa aceptación

Se trata de los errores que puede cometer el usuario al escoger una opción equivocada.

- Confirmación explícita: cuando se le pide al usuario que confirme la acción que se está a punto de hacer. La aplicación no avanza sin un Sí o un No.

Ejemplo: Por favor, si esta correcto digite 1

2.4.6 MÓDULOS DE ASTERISK PARA CONSTRUIR TTS IVR

Existen muchos módulos TTS para Asterisk ya sean libres o con licencia, vamos a describir los más populares.

Festival que es un módulo que viene incorporado dentro de Asterisk este fue desarrollado por Alan W. Black, en la Universidad de Edimburgo distribuida con una licencia BSD como un programa Open Source. Y fue integrado a Asterisk ya que incorpora las ventajas de un TTS sin coste alguno y su desventaja es que su calidad de resultados deja que desear.

Cepstral es el TTS oficial ya que Digium la recomienda al ser partner con Cepstral LLC este software tiene la capacidad de transmitir audio de voz sintetizada, con audios de muy buena calidad y se integra con Asterisk fácilmente, pero tiene que adquirir una licencia ya que no es un software open source.

Google TTS, es un sistema On-Demand de Google, que no cumple con su propósito, este se ofreció como servicio fonético en el sistema de traducción de Google para Google Chrome

2.4.7 TTS IVR CON BASE DE DATOS

Para la integración de nuestro TTS IVR con Base de datos vamos a explicar que es plan de marcado, conectores a base de datos, Asterisk Gateway Interfaz (AGI). Que son las herramientas que utilizamos para la creación de nuestra aplicación.

Plan de marcado que en Asterisk lo conocemos como dialplan, el cual sería como el corazón de nuestro sistema. El cual es similar a un lenguaje script, con funciones, aplicaciones y recursos que se intercalan para formar un programa, y si digito un numero en una terminal recorrerá el dialplan buscando que instrucción va a realizar. Algunos conceptos más importantes que forman nuestro dialplan son contexto, extensiones, prioridad, aplicaciones. Que se encuentran en un fichero llamado extensions.conf el cual lo vamos a localizar en el directorio /etc/Asterisk.

Contexto.- el dial plan se divide en secciones conocidas como contextos, estos son nombres para grupo de extensiones, los cuales contienen una serie de procedimientos para diferentes propósitos. La estructura de un contexto tiene dos partes:

- La parte superior que se escribe el nombre del contexto en corchetes por ejemplo [nombre del contexto]
 - Luego del nombre van las instrucciones que realizara el contexto, cada línea comienza con la palabra **exten** => seguidas de tres valores nombre, prioridad, aplicación.
- ✓ Nombre: normal mente se les conoce como al número que se asigna a los terminales telefónicos llamadas extensiones, estas extensiones en Asterisk no se limita a caracteres numéricos sino también a caracteres alfanuméricos.

exten => 123

- ✓ **Prioridad:** cada extensión puede tener varias funciones o aplicaciones de forma secuencial por lo que se tienen un número de prioridad, el cual se ejecuta según su orden establecido.

`exten => 123,1`

- ✓ **Aplicación:** son funciones predefinidas en Asterisk las cuales tienen diferentes acciones sobre el canal actual como por ejemplo hacer una llamada, colgar, etc.

`exten => 123, 1, Answer ()`

Con estos conceptos vamos a tener una idea más clara de lo que es el dialplan y cómo vamos a crear nuestro TTS IVR ya que esta será un contexto más, un ejemplo de un ivr sería por ejemplo:

[IVR]

`exten => s,1,Wait(1) ;espera un segundo`

`exten => s,2,Set(CHANNEL(language)=es) ; pone como predefinidas las voces en español`

`exten => s,3,Set(TIMEOUT(digit)=7) ; 7 segundos es el tiempo que espera entre el primer dígito y los sucesivos`

`exten => s,4,Set(TIMEOUT(response)=10) ; 10 segundos es el tiempo que espera para que el llamante presione una tecla`

`exten => s,5,BackGround(custom/espeng) ; presenta el menú vocal y al mismo tiempo escucha si el llamante presiona alguna tecla`

`exten => s,6,WaitExten() ; espera que el llamante presione alguna tecla`

`exten => 1,1,goto(IVR1,s,1) ; si presiona 1 va al contexto IVR1, extensión s, prioridad 1`

`exten => 2,1,goto(IVR2,s,1) ; si presiona 2 va al contexto IVR2, extensión s, prioridad 1`

`exten => i,1,Playback(invalid) ; si el número digitado no es válido (ni 1 ni 2) comunica el error`

`exten => i,2,Playback(goodbye) ; se despide`

`exten => i,3,Hangup ; cuelga la llamada`

`exten => t,1,goto(IVR,s,2) ; si dentro de 10 segundos el llamante no presiona ninguna tecla vuelve a presentar el menú vocal`

`exten => h,1,Hangup ; si el llamante cuelga ejecuta la extensión h`

Asterisk Interfaz Gateway en siglas en ingles AGI y en nuestro idioma interfaz pasarela de Asterisk es un protocolo de interfaz de software y comunicaciones.

AGI permite a un programa externo, desarrollado por cualquier usuario, tomar el control del dial plan. Se utilizan para comunicarse con base de datos relacionales, ejecución de archivos, etc.

Los lenguajes más populares para programación de scripts AGI son: PHP, Python y Perl, pero se podría utilizar cualquier otro lenguaje.

El funcionamiento de AGI es el siguiente:

- Primero se establece una pasarela, entre el canal en curso y un scripts ejecutable a voluntad
- Una vez establecida la conexión, se lanza una serie de variables al scripts ejecutando, que son accesibles desde el mismo.
- El script puede ejecutar cualquier algoritmo y entre sus secuencias puede ejecutar comandos específicos de AGI para influenciar el flujo de la llamada. Este script puede seguir en ejecución incluso después de haber finalizado la llamada

El tipo de pasarela que vamos a utilizar es AGI porque también existen EAGI, Fast AGI, Async AGI y estas no las vamos a explicar por qué no es de nuestro interés para el desarrollo de nuestro proyecto.

AGI sería la aplicación estándar, la cual ejecuta la aplicación en la misma máquina que Asterisk, y se abre el flujo de entrada y salida clásico por la entrada y salida asociada correspondiente.

VARIABLES DE ENTRADA una vez establecida la conexión con cualquier pasarela, se lanzan todas las variables a la aplicación y estas son accesibles. Las variables tienen mucho que ver con el canal de uso en curso, por ejemplo

- `agi_request`: nombre del script
- `agi_channel`: nombre del canal
- `agi_language`: lenguaje del canal (en, es)
- `agi_type`: tipo de canal (sip,iax,dahdi)
- etc

PROCESAMIENTO Y COMANDOS una vez que se estableció la pasarela y los valores son asignados es el momento de lanzar los comandos a nuestra necesidad con el fin de interactuar desde fuera de nuestra máquina con Asterisk.

Existen muchos comandos específicos con diversas funcionalidades como por ejemplo:

- `ANSWERS`: responde la llamada entrante
- `CHANNEL STATUS`: vemos el estado de la llamada
- `VERBOSE`: imprime un mensaje en el log
- `WAIT FOR DIGIT`: espera que se presione un dígito
- `[SET | GET] VARIABLE`: asigna u obtiene el valor de una variable del plan de marcado

Entornos de programación considerando que los comandos se pasan a Asterisk por la salida estándar e igual que ocurre con las variables que entran en nuestro script por la entrada, es decir son escalables prácticamente a cualquier sistema o lenguaje de programación que pueda establecer esta comunicación con nuestro sistema UNIX. Para nuestro sistema TTS IVR vamos a escoger PHPAGI que son clases para ser usadas en Asterisk ya que su instalación es simple solo debemos

descargarnos el archivo, y lo descomprimos en el directorio /var/lib asterisk/agi-bin.

Y por último vamos a explicar el conector que vamos a utilizar para hacer uso de nuestra base de datos el cual FreeTDS. Esta es una biblioteca de programación de software libre la que incluye una biblioteca ODBC (es un estándar de acceso a base de datos el objetivo de ODBC, es hacer posible acceder a cualquier dato desde cualquier aplicación y utiliza dos modos, uno como manejador de cliente y cliente servidor), con las cuales permiten a aplicaciones open source como Perl o PHP conectarse a SYBASE o Microsoft SQL SERVER. Algunas características son:

- Es una biblioteca de código fuente y no un programa en sí mismo
- Para su utilización es necesario compilar dicha biblioteca
- Permite que los programas puedan utilizar la API Free TDS a través de enlaces.
- Y está licenciado bajo términos GNU Lesser General Public Licence.
- Se utiliza junto con un módulo para base de datos de ese lenguaje MSSQL SERVER en PHP.

Con todas estas herramientas ya explicadas tendremos una idea del desarrollo de nuestra aplicación.

2.5 HERRAMIENTAS DE DESARROLLO WEB

En el Mercado existe gran infinidad de herramientas para el desarrollo web como son IDEs tal es el caso de dreamweaver, visual studio, etc. así como también existen lenguajes de programación los más utilizados php, asp, perl, Ruby entre otros. Para nuestro caso utilizaremos asp debido a que todo el sitio del municipio de Antonio Ante está desarrollado en esa plataforma. Y en este capítulo daremos una breve revisión lo que es el diseño y maquetación web ya que el sitio ya pasó por esa etapa y

lo que haremos será utilizar la plantilla para realizar nuestra consulta de consumo de agua potable.

2.5.1 DISEÑO Y MAQUETACIÓN

Diseño: En sus inicios la world wide web, los hipertextos se limitaban a presentar un texto impreso segmentado en bloques con algunos hipervínculos, esto ha cambiado, en la actualidad esto sea a convertido en obras de arte de diseño gráfico, multimedia, integración con base de datos, servicios online, etc. en el cual ya no solo importa su contenido si no también su diseño y funcionalidad.

Se pueden establecer las siguientes etapas para el diseño Web:

- **Delimitación del tema:** de qué va a tratar el sitio web.
- **Recolección de la información:** recopilar y seleccionar la información que se va a incluir.
- **Agregación:** hacer un balance equilibrado entre linealidad y jerarquización.
- **Estructuración de los contenidos:** En esta etapa se definen tanto las estructuras jerárquicas y horizontales, como las taxonomías y esquemas de clasificación.
- **Creación de los sistemas de navegación y búsqueda:** creación de páginas guía, ayudas a la navegación, tablas de contenido, índices, sumarios, mapas de navegación, glosarios, páginas de búsqueda, uso de iconos y barras de navegación, utilización de metáforas, etc. accesibles desde cualquier otra página del sitio web.
- **Diseño y estilo gráfico:** estilos y formatos textuales, coherencia gráfica, diseño de fondos y distribución de los elementos dentro de la página, inclusión de material multimedia, cantidad y tamaño de las imágenes, etc. Dar homogeneidad y coherencia a todo el sitio web.

- **Ensamblaje final:** últimos enlaces, diseño de portadas y estilos gráficos, logotipos, enlaces sobre autoría, contacto, fechas de creación o de actualizaciones, etc.
- **Evaluación y test de uso:** comprobación del funcionamiento, vínculos y páginas rotas, usabilidad, accesibilidad ajustes, etc.

Maquetación: esta consta de dos etapas:

- La primera etapa se realiza solo con HTML y se trata de crear una estructura de etiquetas para albergar todo el contenido y las distintas secciones de nuestra página web.
- La segunda se realiza con CSS, y consiste en darle a la página el aspecto visual y determinar la colocación en la pantalla de todos sus elementos.

Existe un paso antes de realizar la maquetación que se denomina wireframing consiste en realizar un boceto sencillo de lo que queremos hacer en la página web. Este se lo puede realizar a mano o con alguna herramienta online.

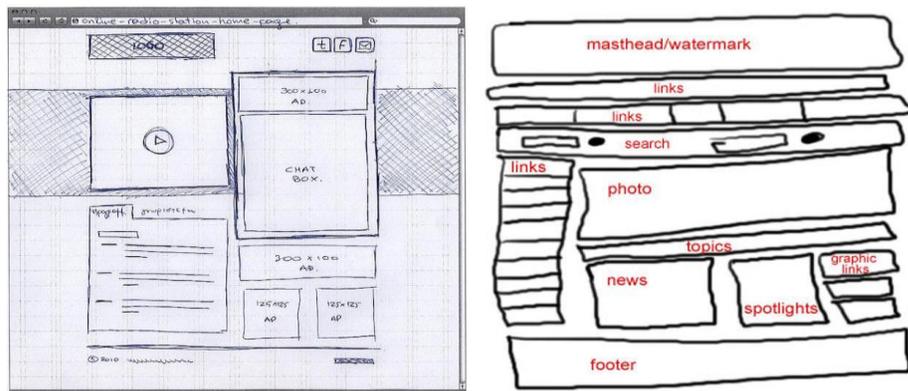


Figura 9: Grafica de diseño y maquetación Fuente: <http://josejfernandez.com/categoria/desarrollo-web>

2.5.2 DESARROLLO WEB EN PHP y ASP.NET

Vamos hablar de estos dos lenguajes de programación web como es php el cual vamos a utilizar en nuestra aplicación TTS IVR la cual va a hacer la interacción entre Asterisk y la base de datos y asp.net el cual vamos a utilizar en el portal web debido a que la plataforma del Gobierno Autónomo Descentralizado de Antonio Ante esta basado en esa plataforma.

PHP (acrónimo recursivo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Bien, pero ¿qué significa realmente? Un ejemplo nos aclarará las cosas:

Ejemplo #1 Un ejemplo introductorio

```
<!DOCTYPE HTML PUBLIC "-
//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>
    <?php
      echo "¡Hola, soy un script de PHP!";
    ?>
  </body>
</html>
```

En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!"). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final `<?php?>` que permiten entrar y salir del "modo PHP".

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.

Aunque el desarrollo de PHP está centrado en la programación de scripts del lado del servidor, se puede utilizar para muchas otras cosas. (php.net, 2015)

Con php podemos hacer cualquier cosa, ya que está orientado a la programación de scripts del lado del servidor, como son recopilar datos de formularios, general páginas dinámicas entre muchas cosas más.

Donde se usa scripts php:

- Scripts del lado del servidor. Para esto necesita tres elementos el analizador de PHP (servidor), un servidor web y un navegador. Esto se puede ejecutar en una máquina y poder visualizar el resultado en un navegador.
- Scripts desde la línea de comandos. Se puede ejecutar directamente un script PHP en consola linux sin la necesidad de un navegador.
- Escribir aplicaciones de escritorio. Php utiliza características avanzadas del lado del cliente PHP-GTK para crear estas aplicaciones.

(php.net, 2015)

ASP.NET es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible

con el Common Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia, etc.

ASP.NET incluye:

- Marco de trabajo de página y controles
- Compilador de ASP.NET
- Infraestructura de seguridad
- Funciones de administración de estado
- Configuración de la aplicación
- Supervisión de estado y características de rendimiento
- Capacidad de depuración
- Marco de trabajo de servicios Web XML
- Entorno de host extensible y administración del ciclo de vida de las aplicaciones
 - Entorno de diseñador extensible. (msdn.microsoft.com, 2007)

Como lenguaje vamos a escoger C#

C# es un lenguaje orientado a objetos elegante y con seguridad de tipos que permite a los desarrolladores compilar diversas aplicaciones sólidas y seguras que se ejecutan en .NET Framework. Puede utilizar C# para crear aplicaciones cliente de Windows, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos, y mucho, mucho más. Visual C# proporciona un editor de código avanzado, cómodos diseñadores de interfaz de usuario, depurador integrado y numerosas herramientas más para facilitar el desarrollo de aplicaciones basadas el lenguaje C# y .NET Framework. (msdn.microsoft.com, 2015)

3 IMPLEMENTACION

Un factor muy importante para el desarrollo de nuestro proyecto de titulación es la base de datos, de la cual vamos a obtener toda la información requerida por los usuarios.

El Gobierno Autónomo descentralizado de Antonio Ante cuenta con una base de datos SQL Server 2000, la cual cuenta 250 tablas normalizadas, en las cuales guardan distintos tipos de información.

Para nuestro desarrollo de nuestro proyecto vamos a utilizar una sola tabla la cual nos facilitara el trabajo ya que cuenta con los datos necesarios para realizar nuestra consulta, la cual se denomina Datos_ingreso.

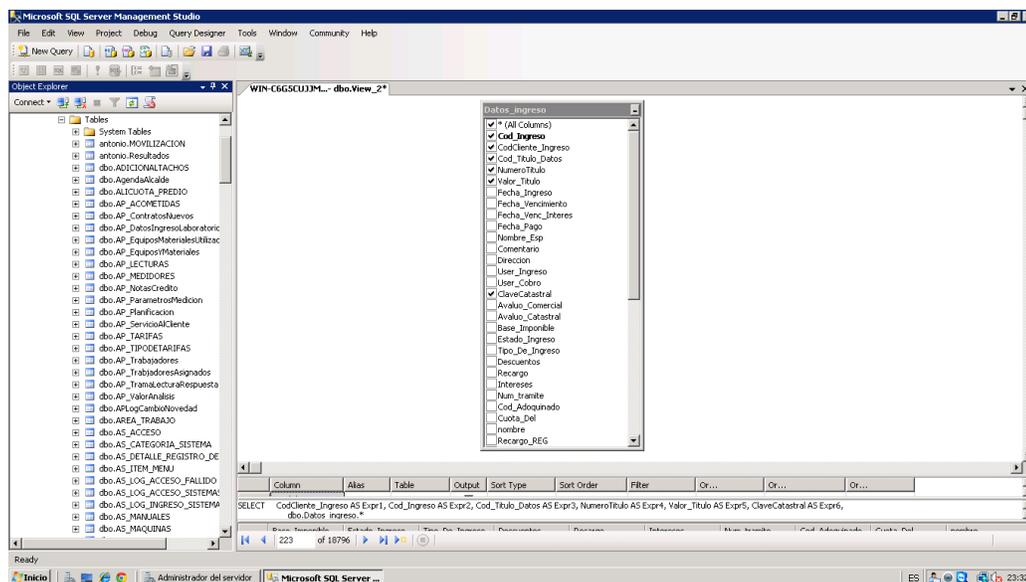


Figura 10: Tabla con la cual realizamos las consultas Fuente: Propia

3.1 INSTALACIÓN DE LIBRERÍAS BASE PARA ASTERISK

Antes de la instalación de las librerías vamos a instalar nuestro Sistema base que será la distribución Centos 6.5 mínima

Antes de empezar la instalación debemos tener la iso y tenerla en un medio óptico, de la distribución de Centos 6.5 que es con la cual vamos a trabajar ya que la instalación de Asterisk la podemos hacer también en otras distribuciones de Linux como es debían, Ubuntu, etc.

Cómo ya sabemos debemos verificar q el arranque del servidor que vamos a utilizar par a nuestra implementación lo haga desde el medio óptico. Una vez que arranco desde la distribución de Centos nos presenta la siguiente pantalla



Figura 11: Inicio de instalación Centos Fuente: Propia

Y damos entre en instalación

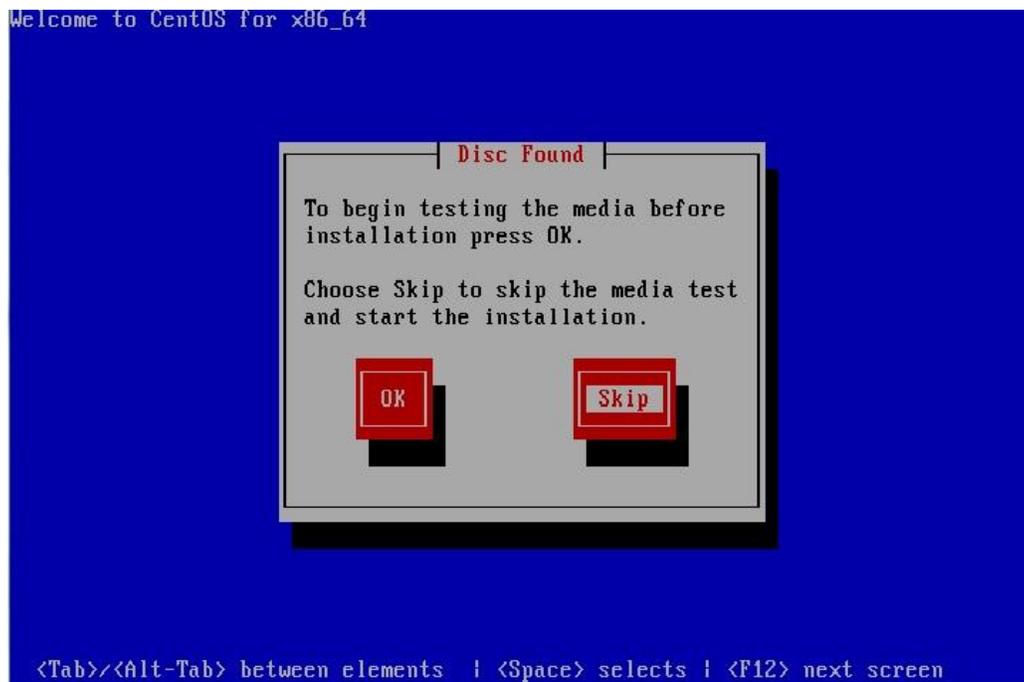


Figura 12: verificación de cd de instalación Fuente: Propia

En esta pantalla nos indica si queremos hacer un test de nuestro medio óptico pero saltamos este paso poniendo en omitir



Figura 13: Logo de inicio de instalación Centos Fuente: Propia

En esta pantalla comenzamos con la instalación y damos clic en siguiente

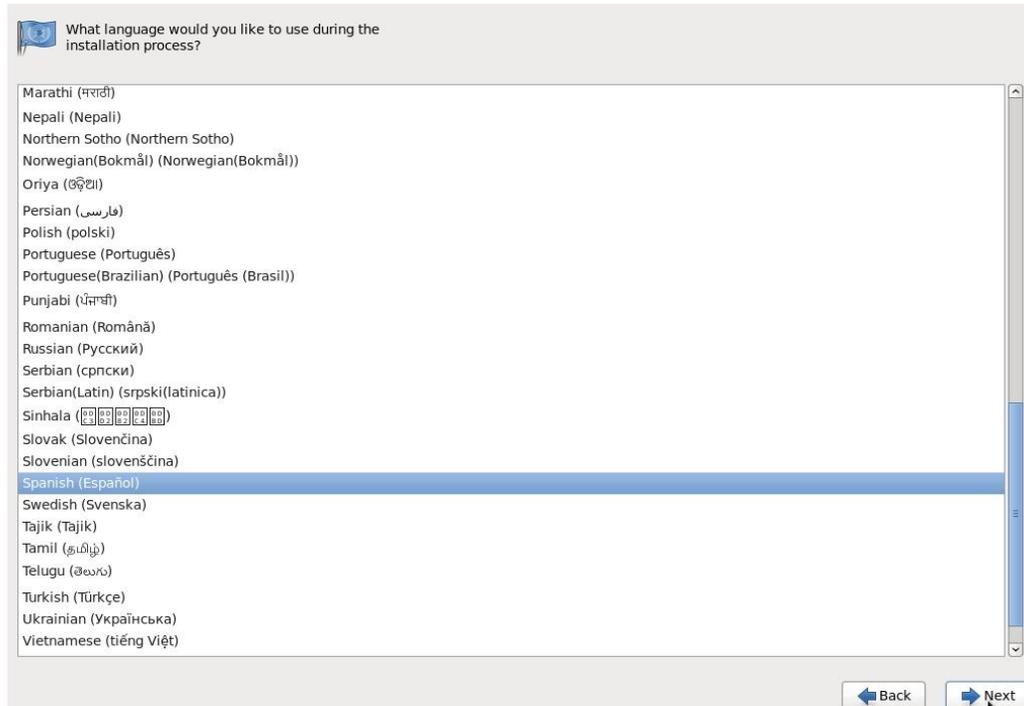


Figura 14: Selección de idioma de instalación de Centos Fuente: Propia

En esta pantalla escogemos el idioma de la instalación y damos clic en siguiente

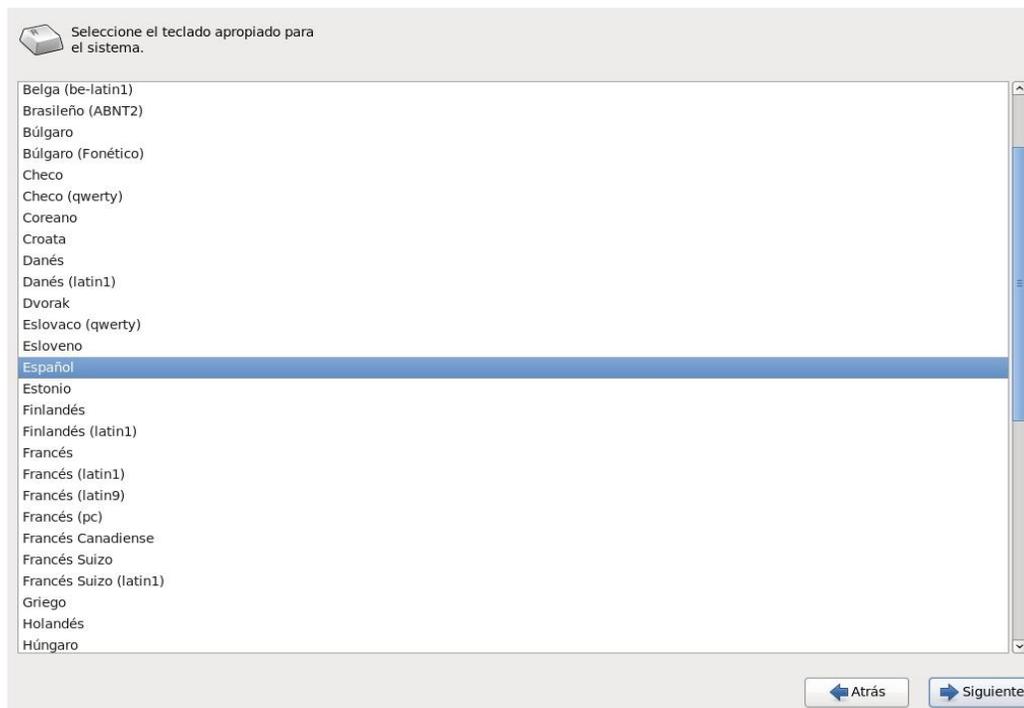


Figura 15: Selección de idioma del teclado Fuente: Propia

En esta pantalla escogemos el idioma del teclado para nuestra instalación escogemos en español y damos clic en siguiente



Figura 16: Dispositivos de almacenamiento para la instalación de Centos Fuente: Propia

En esta pantalla nos dice en qué tipo de discos tenemos para nuestra instalación activamos en dispositivos básicos ya que contamos con un único disco duro

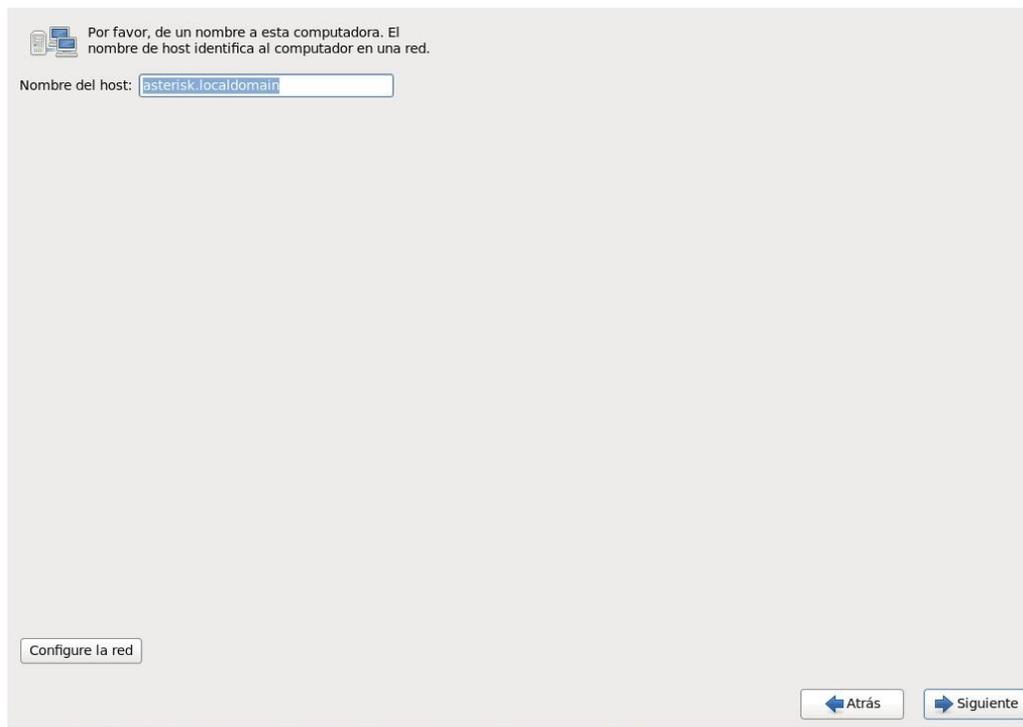


Figura 17: Nombre de nuestro host Fuente: Propia

En esta pantalla escribimos el nombre de nuestra máquina y dominio además configuramos la tarjeta de red si lo deseamos y damos clic en siguiente

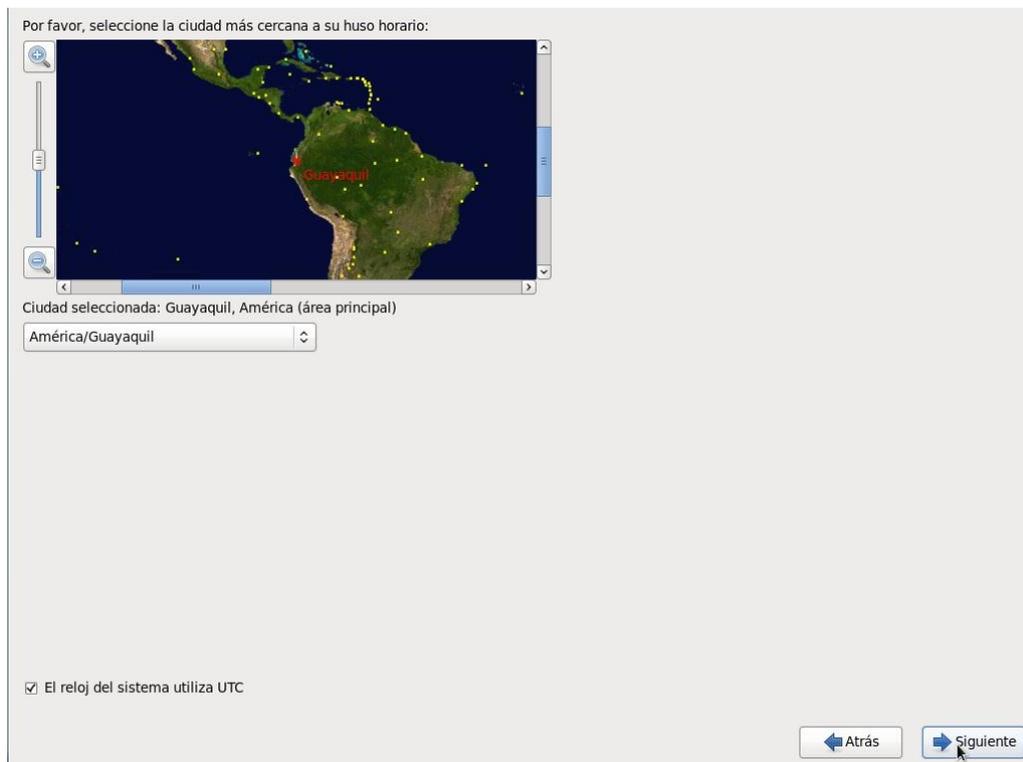
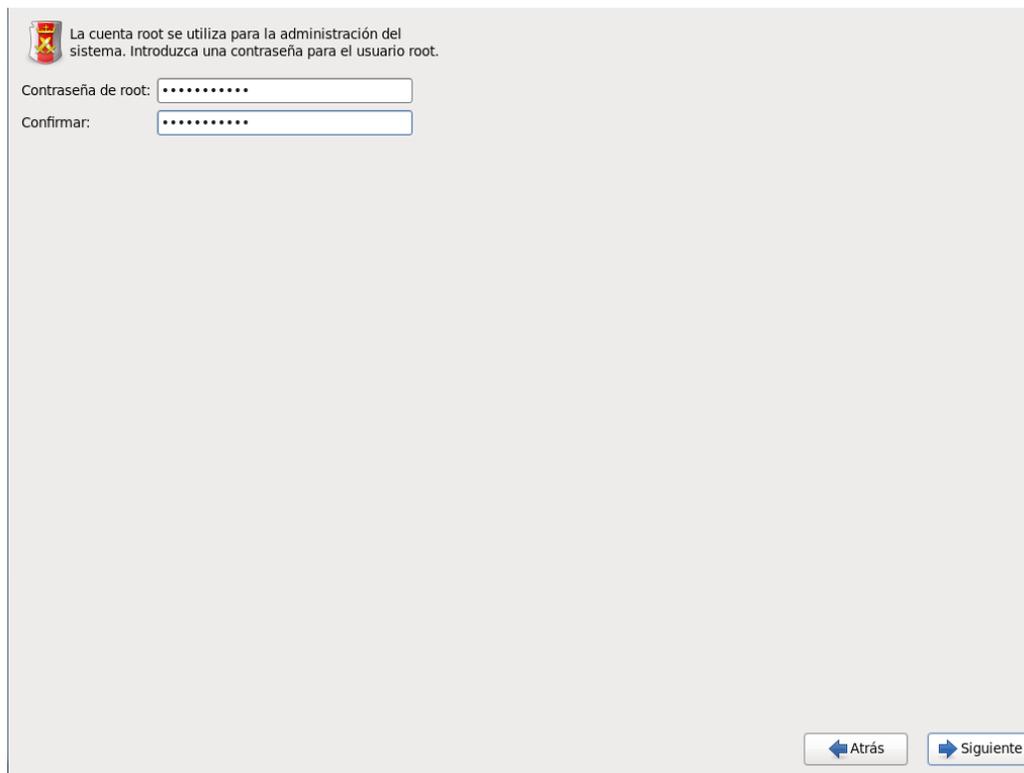


Figura 18: Indicamos Nuestra Ubicación de nuestro País Fuente: Propia

En esta pantalla colocamos nuestra zona horaria para nuestro caso será Guayaquil Ecuador y damos clic en siguiente



La cuenta root se utiliza para la administración del sistema. Introduzca una contraseña para el usuario root.

Contraseña de root:

Confirmar:

[← Atrás](#) [→ Siguiente](#)

Figura 19: Digitamos nuestra contraseña de súper usuario Fuente: Propia

En esta pantalla escribimos la contraseña del súper usuario root y damos clic en siguiente

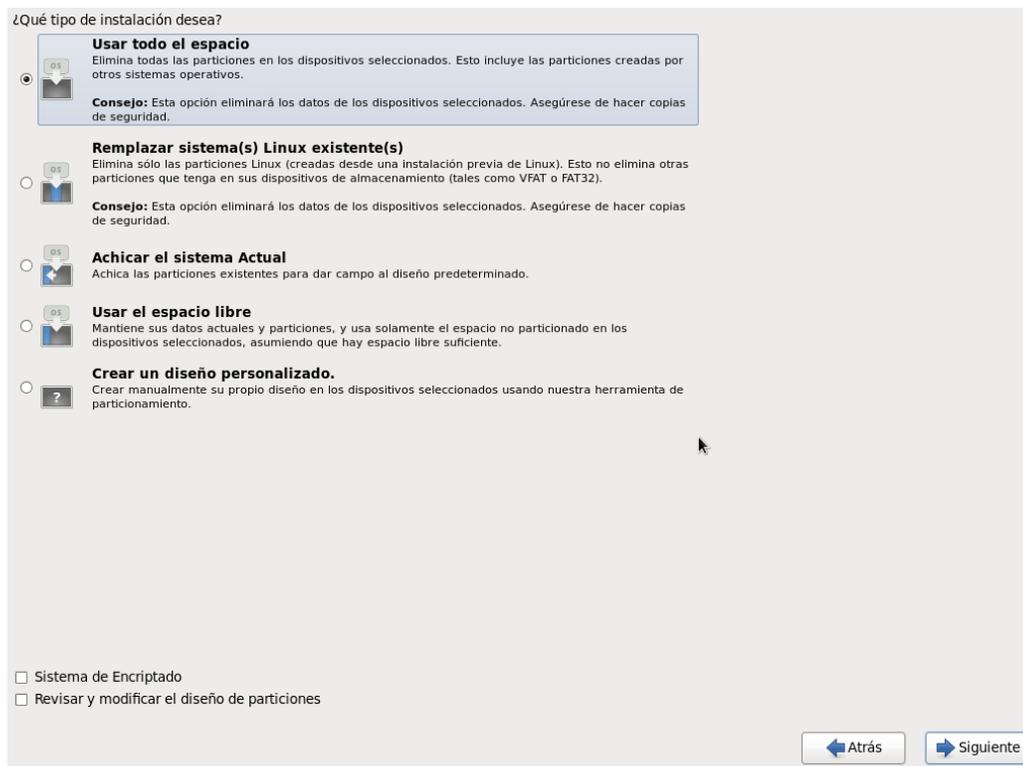


Figura 20: Indica el tipo de instalación en nuestro disco Fuente: Propia

En esta pantalla activamos donde dice usar todo el espacio ya que vamos a utilizar todo el disco duro y damos clic en siguiente

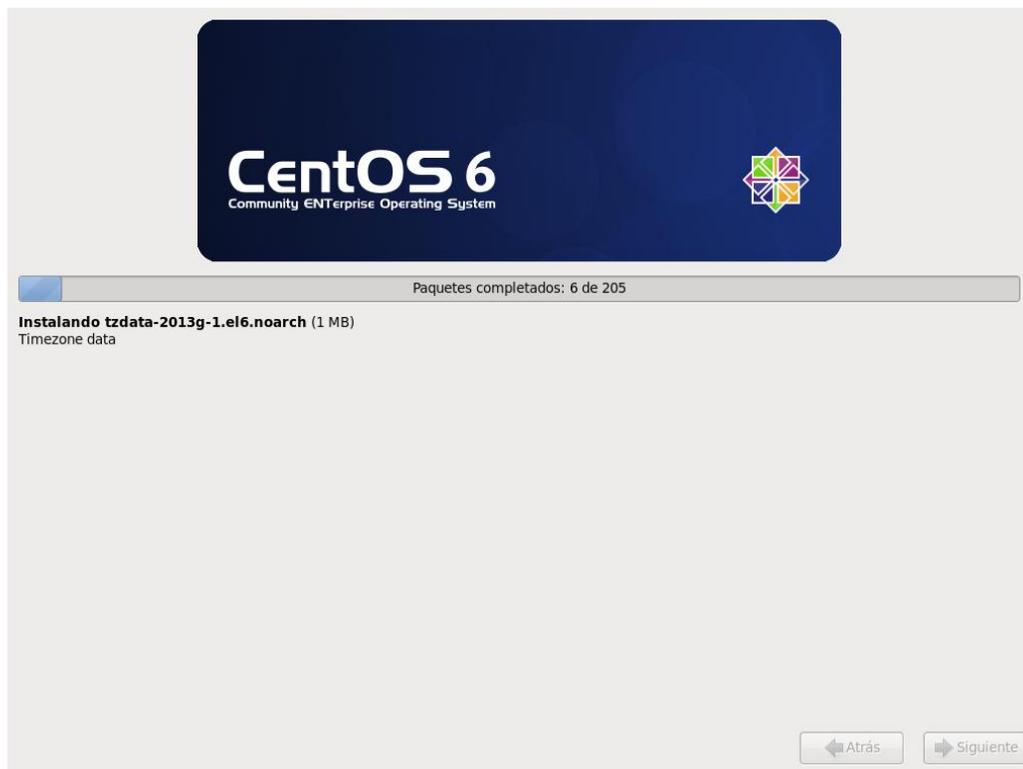


Figura 21: Proceso que inicio a instalar todos los paquetes del sistema Fuente: propia

En esta pantalla se está instalando la distribución de Centos

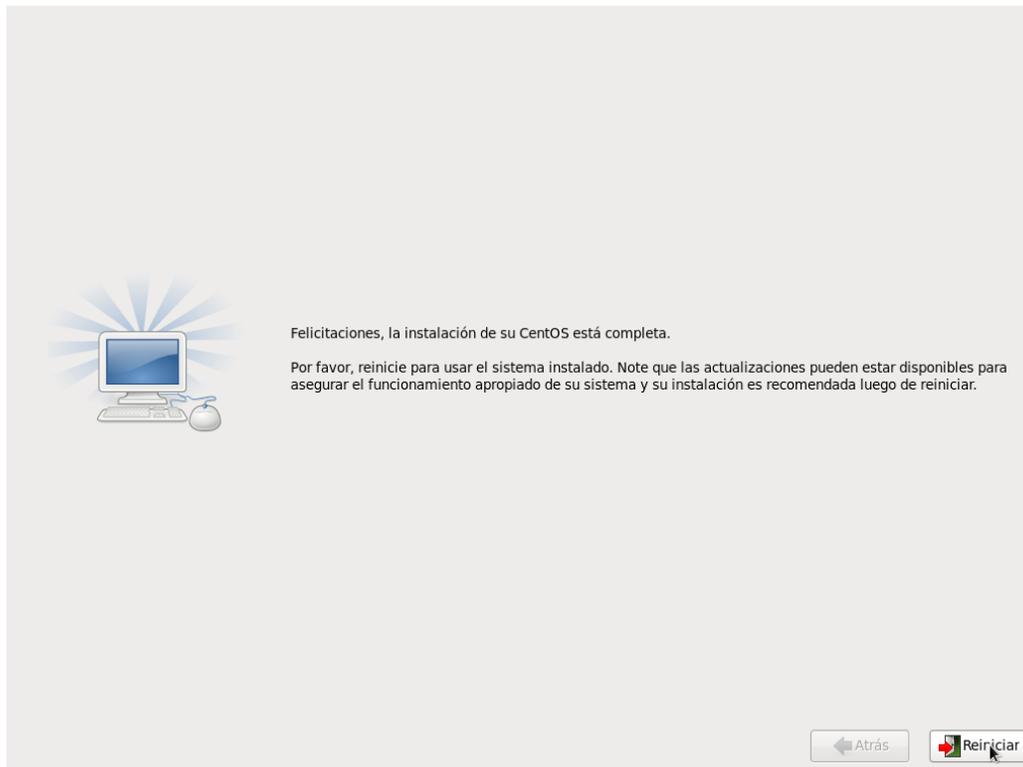


Figura 22: La instalación de Centos ha finalizado Fuente: Propia

Una vez terminada la instalación nos indicara que debemos reiniciar la máquina y damos clic en reiniciar

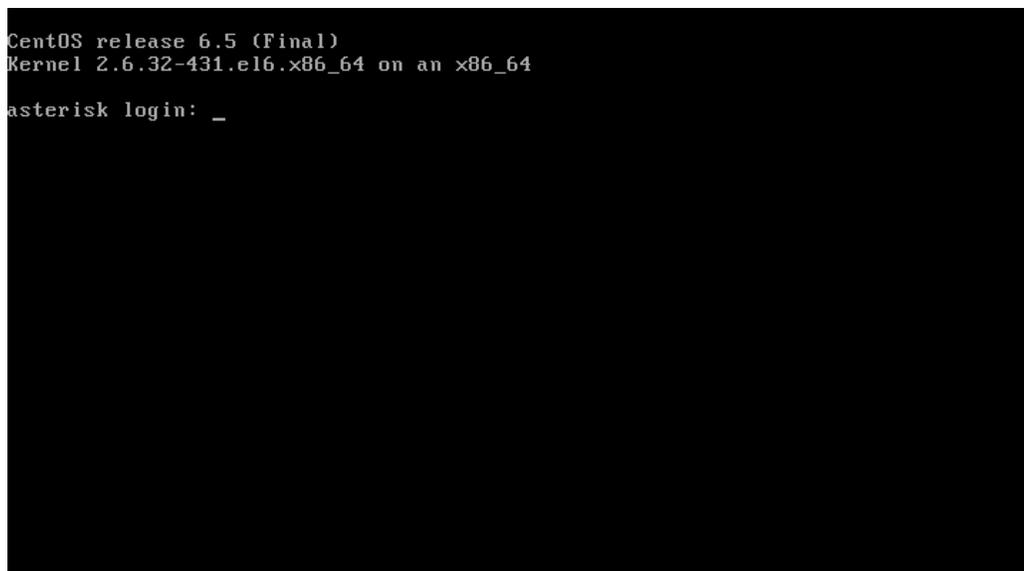


Figura 23: Primer inicio de Centos Fuente: Propia

Una vez que la maquina se ha reiniciado nos aparece esta pantalla en la cual digitamos el usuario y la contraseña

```
CentOS release 6.5 (Final)
Kernel 2.6.32-431.el6.x86_64 on an x86_64

asterisk login: root
Password:
[root@asterisk ~]#
```

Figura 24: Nos pide loguearse para ingresar al sistema Fuente: Propia

Una vez ingresado en Centos vamos a configurar la tarjeta de red para poder instalar las dependencias, librerías necesarias para la instalación de Asterisk ya que no la configure anteriormente.

```
CentOS release 6.5 (Final)
Kernel 2.6.32-431.el6.x86_64 on an x86_64

asterisk login: root
Password:
[root@asterisk ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

Figura 25: Instrucción de para configurar la tarjeta de red Fuente: Propia

En esta pantalla vamos a modificar el archivo de configuración de nuestra tarjeta de red


```

DNS1=200.105.225.2
DNS2=200.105.225.4

"/etc/sysconfig/network-scripts/ifcfg-eth0" 12L, 239C written
[root@asterisk ~]# /etc/init.d/network restart
Interrupción de la interfaz de loopback:           [ OK ]
Activación de la interfaz de loopback:           [ OK ]
Activando interfaz eth0: Determining if ip address 192.168.2.73 is already in use for device eth0...
                                                    [ OK ]

[root@asterisk ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=49 time=76.7 ms

```

Figura 28: Ping de prueba de salida a internet Fuente: Propia

Una vez comprobado que ya tenemos salida a internet procedemos a instalar las dependencias necesarias para la instalación de Asterisk ya que nuestra distribución es la versión mínima.

Para esto vamos a instalar:

Vim que es un editor de texto o el editor de texto de nuestra preferencia.

Wget que nos servirá para descargar nuestros paquetes

Librerías son requisitos imprescindibles para la instalación de Asterisk:

1. libssl-dev: Librería necesaria para gestionar tramas cifradas.
2. Zlib1g-dev: Librería para comprimir y descomprimir datos mediante el algoritmo zlib.
3. Libncurses5-dev: Librería para utilizar el modo pseudo-grafico en la consola de texto.
4. libnewt-dev: Librería para mostrar un interfaz de color en la consola de texto.
5. cerner headers o kernel sources: Los archivos *includes* o bien el código fuente del kernel de Linux que estemos utilizando.

Aplicaciones necesarias:

1. gcc : Compilador de C
2. g++ : Compilador de C++
3. make: Automatizador para la compilación rápida y fácil.

Una vez que ya está instalado Centos siempre es recomendable actualizarlo de la siguiente forma En nuestra consola digitamos:

```
#yum -y update
```

```
#yum groupinstall core
```

```
#yum groupinstall base
```

Instalamos dependencias adicionales requeridas

```
# yum install gcc gcc-c++ lynx bison mysql-devel mysql-server php php-mysql php-pear php-mbstring tftp-server httpd make ncurses-devel libtermcap-devel sendmail sendmail-cf caching-nameserver sox newt-devel libxml2-devel libtiff-devel audiofile-devel gtk2-devel subversion kernel-devel git subversion kernel-devel php-process crontabs cronie cronie-anacron wget vim php-xml uuid-devel libtool sqlite-devel
```

Instalar los paquetes necesarios CDR ODBC

```
# yum install libtool-ltdl-devel unixODBC unixODBC-devel mysql-connector-odbc
```

Compruebe si el firewall (iptables) está activada por defecto.

Ejecutar system-config-firewall-tui y configurar el firewall según sea necesario.

Como mínimo, los siguientes puertos deben estar abiertos:

```

system-config-firewall-tui
TCP 80 (www)
TCP 4445 (Flash Operator Panel)
UDP 5060-5061 (SIP)
UDP 10,000 - 20,000 (RTP)
? UDP 4569 (IAX)

```

Como alternativa se puede desactivar el firewall e impedir que se active cuando se reinicie la máquina de la siguiente manera:

```

# Service iptables stop
# chkconfig iptables off

```

Se Linux

En el archivo de configuración vamos a deshabilitar cambiando en la línea

```
SELINUX= enforcing
```

Por

```
SELINUX=disabled
```

```
vim /etc/selinux/config
```

```

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

```

Figura 29: Archivo de configuración Selinux módulo de seguridad Fuente: Propia

TFTP

Si vamos utilizar teléfonos ip por hardware tendremos que habilitar esta opción para aprovisionamiento.

```
vim /etc/xinetd.d/tftp
```

Cambiamos

```
server_args = -s /var/lib/tftpboot
```

Por:

```
server_args = -s /tftpboot
```

Cambiamos

```
disable=yes
```

Por:

```
disable=no
```

```
# default: off
# description: The tftp server serves files using the trivial file transfer \
# protocol. The tftp protocol is often used to boot diskless \
# workstations, download configuration files to network-aware printers, \
# and to start the installation process for some operating systems.
service tftp
{
    socket_type      = dgram
    protocol        = udp
    wait            = yes
    user            = root
    server          = /usr/sbin/in.tftpd
    server_args     = -s /tftpboot
    disable         = no
    per_source      = 11
    cps             = 100 2
    flags           = IPv4
}
```

Figura 30: Archivo de configuración tftp para aprovisionamiento Fuente: Propia

Luego digitamos para crear la carpeta tftpboot y otorgamos permisos para lectura y escritura

```
mkdir /tftpboot
```

```
chmod 777 /tftpboot
```

Por ultimo reiniciamos el servicio xinetd

```
service xinetd restart
```

Colocamos nuestra zona horaria

Con el comando `tzselect` o con la línea siguiente:

```
# ln -sf /usr/share/zoneinfo/America/Guayaquil /etc/localtime
```

Verificamos la hora de nuestro sistema con el comando

```
# hwclock
```

A continuación vamos a proceder a la instalación de Asterisk para ello vamos al directorio:

`/usr/src` donde vamos a descargar todos los paquetes a utilizar.

Instalación de lame

LAME (Lame *ain't an mp3 encoder*) se usa con la mayoría de programas que convierten archivos WAV a archivos MP3 o desde otros formatos o soportes.

Descargamos el paquete

```
# wget http://downloads.sourceforge.net/project/lame/lame/3.98.4/lame-3.98.4.tar.gz?ts=1292626574&use\_mirror=cdnetworks-us-1
```

Descomprimos el instalador

```
# tar -xf lame-3.98.4.tar.gz
```

Entramos en el directorio

```
# cd lame-3.98.4
```

Y procedemos a instalar

```
# ./configure --prefix=/usr && make && make install
```

Instalación de Speex

Speex diseñado para comprimir voz a bitrates desde 2 a 44 kbps y posee características que no tiene otros códec de voz como codificación de intensidad estéreo

Descargamos el instalador

```
# wget http://downloads.xiph.org/releases/speex/speex-1.2rc1.tar.gz
```

Descomprimos

```
# tar -xf speex-1.2rc1.tar.gz
```

Entramos en la carpeta

```
# cd speex-1.2rc1
```

Instalamos

```
# ./configure --prefix=/usr && make && make install
```

Instalación mpg123

Es un reproductor de música en consola

Descargamos el instalador

```
# wget http://sourceforge.net/projects/mpg123/files/mpg123/1.14.2/mpg123-1.14.2.tar.bz2
```

```
# tar -xf mpg123-1.22.0.tar.bz2
```

Ingresamos a la carpeta comprimida

```
# cd mpg123-1.22.0
```

Instalamos

```
# ./configure && make && make install
```

Instalamos phpmyadmin

phpMyAdmin es una herramienta escrita en php con la intención de manejar la administración de MySQL a través de páginas web, se puede crear y eliminar base de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos

Para instalar phpMyAdmin ejecutamos el siguiente comando

```
# yum install phpmyadmin
```

Una vez hecho vamos a modificar el archivo de configuración para poder acceder

desde cualquier ip a la administración. Con los siguientes comandos:

```
vi /etc/httpd/conf.d/phpMyAdmin.conf
```

```
#
```

```
# Web application to manage MySQL
```

```
#
```

```
#<Directory "/usr/share/phpmyadmin">
```

```
# Order Deny, Allow
```

```
# Deny from all
```

```
# Allow from 127.0.0.1
```

Instalación de DAHDI

Dahdi es un paquete de controladores para tarjetas análogas con puertos fxo y fxs

Descargamos el instalador

```
# wget http://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-  
linux-complete-current.tar.gz
```

Descomprimos

```
# tar zxvf dahdi-linux-complete*
```

Entramos en la carpeta descomprimida

```
# cd /dahdi-linux-complete*
```

Procedemos a instalar

```
# make && make install && make config
```

Instalación de libre

```
wgethttp://downloads.asterisk.org/pub/telephony/libpri/libpri-1.4.14.tar.gz
```

```
# tar -xf libpri-1.4.14.tar.gz
```

```
# cd libpri-1.4.14
```

```
# make clean && make && make install
```


Descomprimos el archivo, ingresamos al directorio y verificamos que mysql este levantado, una vez verificado procedemos a crear nuestras bases de datos con el siguiente comando

```
# mysqladmin create asterisk  
# mysqladmin create asteriskcdrdb
```

Y llénanos la base con los script que ya bien en la carpeta de FreePBX con los siguientes comandos

```
mysql asterisk < SQL/newinstall.sql  
mysql asteriskcdrdb < SQL/cdr_mysql_table.sql
```

Damos permisos a nuestro usuario con la siguiente línea de comandos

```
mysql> GRANT ALL PRIVILEGES ON asterisk.* TO asterisk@localhost  
IDENTIFIED BY 'asterisk';
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> GRANT ALL PRIVILEGES ON asteriskcdrdb.* TO asterisk@localhost  
IDENTIFIED BY 'asterisk';
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> flush privileges;
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> \q
```

Una vez hecho esto procedemos a la instalación de FreePBX

Como nuestro sistema operativo es de 64 bits utilizaremos la siguiente línea:

```
#. /install_amp --username=asterisk --password=qazQ18Ag;
```

Procedemos a ejecutar el script de la instalación de FreePBX y ponemos enter en todas las preguntas

Una vez terminada la instalación procedemos a verificar si nuestra frontal de administración está funcionando digitando lo siguiente en cualquier navegador

<http://192.168.34.254/admin/config.php>

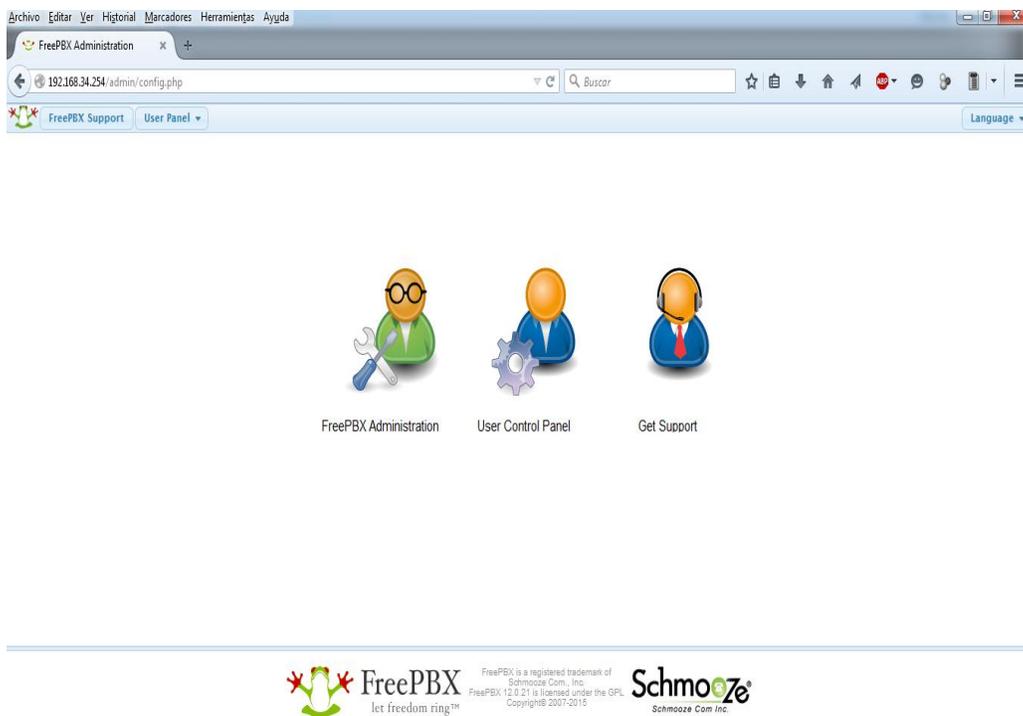


Figura 33: Pantalla principal de FreePBX Fuente: Propia

Esta es la pantalla inicial que nos aparecerá en la cual vamos a loguearnos

3.4 INSTALACIÓN DE FESTIVAL

Como ya hemos hecho una pequeña introducción de festival procederemos con la instalación digitando el comando:

```
# yum install festival festival-devel
```

Una vez instalado verificamos si está funcionando ingresando a la consola de asterisk digitando

```
# asterisk -rvvv
```

```
CLI>module unload app_festival
```

```
== unregistered application 'festival'
```

```
CLI> module load app_festival
```

```
==parsing '/etc/asterisk/festival.conf': ==found
```

```
== Registered application "festival;"
```

Ahora vamos a predefinir el idioma ya que este software viene por defecto en ingles

Primero tenemos que descargarnos las voces en español

```
Cd /usr/share/festival/voices
```

```
wget http://www.voztovoice.org/tmp/festival-spanish.zip
```

Procedemos a extraer

```
unzip festival-spanish.zip
```

Y por último modificamos el archive de configuración de festival

```
nano /usr/share/festival/festival.scn
```

Añadimos estas líneas

```
;(language__spines)
```

```
(set! voice_default 'voice_el_diphone)
```

```
(define (tts_textasterisk string mode)
```

```
"(tts_textasterisk STRING MODE)
```

Apply tts to STRING. This function is specifically designed for use in server mode so a single function call may synthesize the string.

This function name may be added to the server safe functions."

```
(let ((wholeutt (utt.synth (eval (list 'Utterance "Text string")))))
```

```
(utt.wave.resample wholeutt 8000)
```

```
(utt.wave.rescale wholeutt 5)
```

```
(utt.send.wave.client wholeutt)))
```

Guardamos y salimos

3.5 CONFIGURACIÓN DE FREEPBX COMO CENTRAL TELEFONICA

FreePBX es un frontal web para asterisk, con el cual podemos configurar y administrar nuestra centralita haciéndolo de una forma más sencilla ya que tiene sus módulos desarrollados como son: condiciones de tiempo, IVR, creación de extensiones, colas, etc. Estos módulos pueden encontrarse de forma gratuita o de pago ya que existen gran cantidad de aplicaciones para asterisk, también debemos tomar en cuenta que la configuración lo podemos hacer directamente en los archivos de configuración de asterisk esto dependerá de la capacidad del usuario ya sea un principiante o un experto en el manejo de asterisk.

En esta parte de nuestro sistema vamos a dar un valor agregado a nuestro proyecto ya que explicare, las funciones básicas para el funcionamiento de asterisk como una típica central telefónica. Como son:

Creación de extensiones, dialplan, configuración de tarjetas FXO, IVR, creación de rutas entrantes y salientes, etc.

Antes de adentrarnos en nuestra interfaz explicare brevemente de que consta, una vez que ingresamos al sistema nos muestra una pantalla con la información del sistema el cual la veremos así:

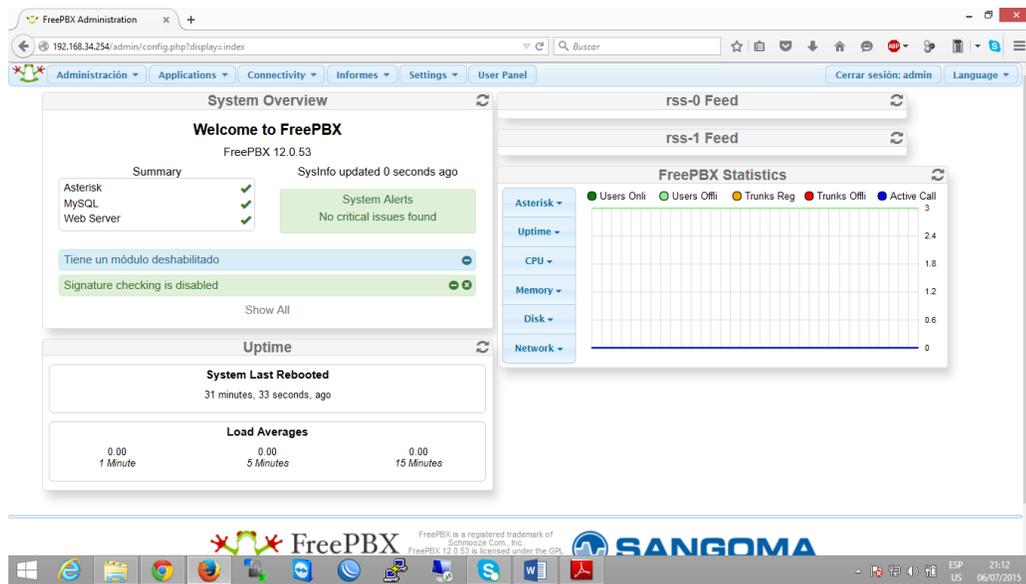


Figura 34: Estado del Sistema Fuente: Propia

La cual nos dice el estado de asterisk, mysql y nuestro servidor web que debe estar ok, además nos presenta información del estado de encendido del servidor, consumo de recursos del CPU, memoria, disco duro y de red.

Hacemos clic en la pestaña de administración en la cual nos presenta herramientas para administrar asterisk como son copias de seguridad, administración de módulos, grabaciones del sistema los cuales utilizaremos para dar anuncios en nuestro IVR, etc. Estas herramientas serán de uso dependiente a los requerimientos de cada necesidad.

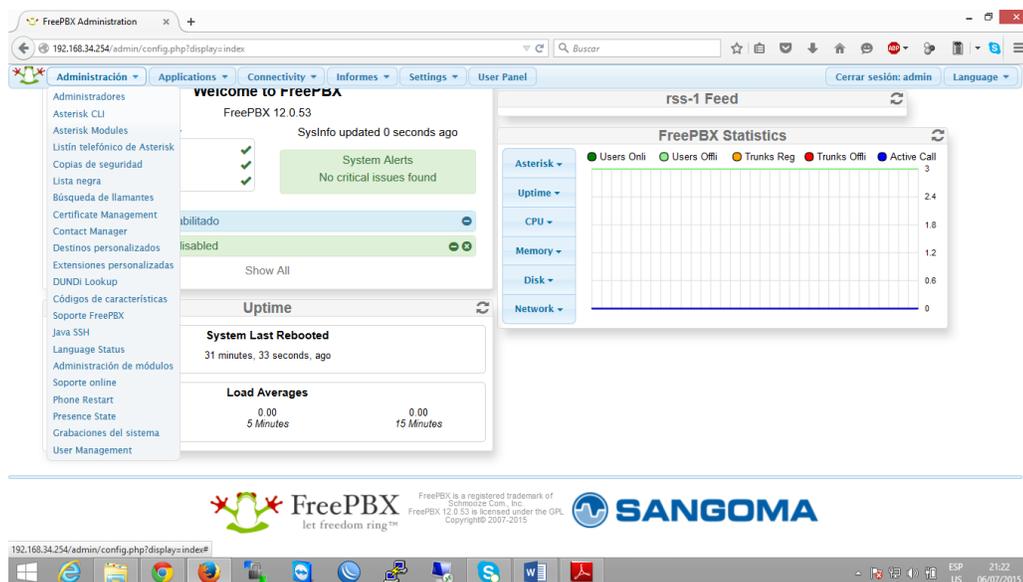


Figura 35: Herramientas de administración Fuente: Propia

Nos dirigimos a la pestaña de aplicaciones en la cual encontramos diferentes módulos como son creación de extensiones, grabación de llamadas, intercomunicador, colas, grupos horarios, etc .

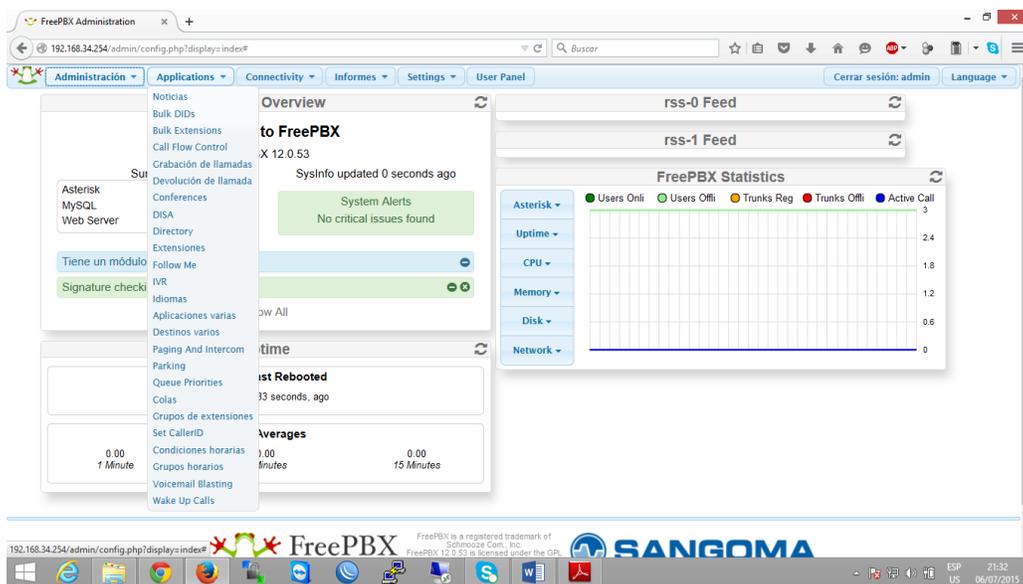


Figura 36: Aplicaciones para asterisk Fuente: Propia

Conectividad en esta pestaña encontramos herramientas para configurar nuestra tarjeta que será la encargada de recibir y realizar llamadas hacia la red telefónica tradicional los módulos a destacar son: dahdi config, dahdi channel DIDS.

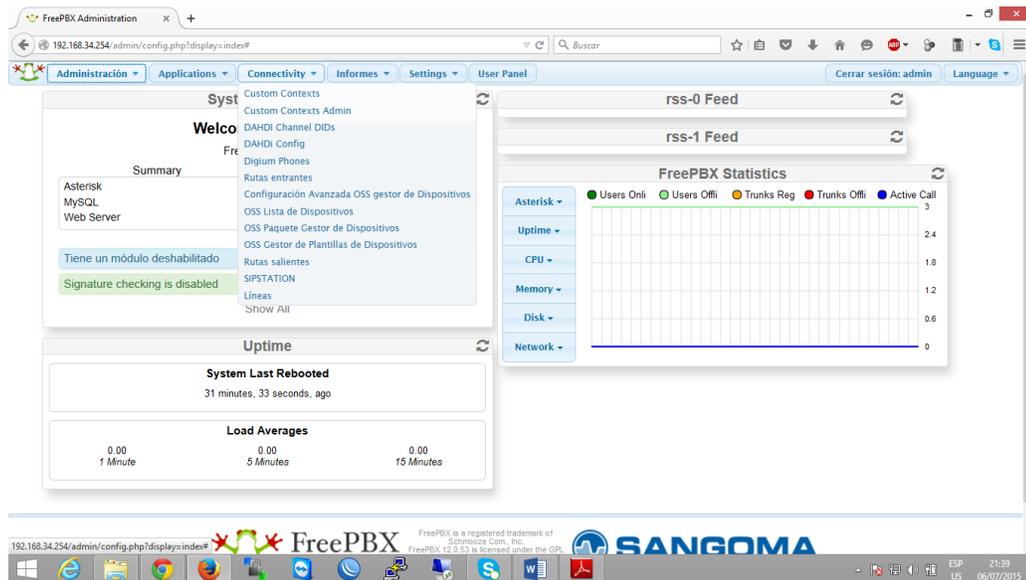


Figura 37: Configuraciones Fuente: Propia

Por ultimo vamos a revisar la pestaña de configuraciones la cual cuenta con diferentes tipos de configuraciones ya sea para SIP, IAX2, extensiones, música en espera, etc.

Una vez explicado rápidamente FreePBX vamos a configurar las extensiones, para eso vamos a aplicaciones y hacemos clic en extensiones, no dirá que seleccionemos el tipo de dispositivo y elegimos SIP ya que hay muchos dispositivos que cuentan con este protocolo.



Figura 38: Anadir extensión Fuente: Propia

También existen dispositivos con protocolo IAX2 pero en nuestro mercado todavía no lo distribuyen, una vez escogido la tecnología del dispositivo damos clic en enviar presentándonos la siguiente pantalla:

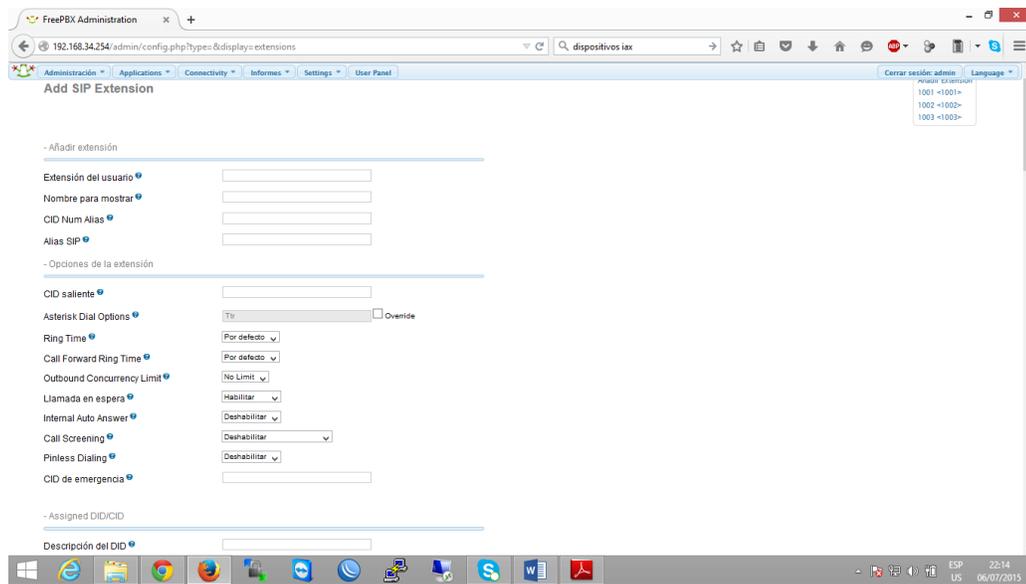


Figura 39: Creación de una extensión Fuente: Propia

Existen muchas opciones que nos presenta en pantalla pero solo basta llenar unas pocas para su funcionamiento las más importantes son:

Extensión del usuario.- Es un número único que sirve para identificar una extensión telefónica.

Nombre para mostrar.- aquí digitaremos el nombre o etiqueta que se mostrara cuando hacemos una llamada a otra extensión.

Ring time.- es el tiempo que se demora asta enviarlo al Voice mail o colgar la llamada.

Llamada en espera.- como su nombre lo indica en esta opción habilitamos esta opción.

Secret.- ingresamos una contraseña para que el dispositivo se autentifique y registre la extensión con nuestra central asterisk.

DTMF Signaling.- especifica qué tipo de tono DTMF va a esperar la central asterisk generalmente se elige rfc2833 que es la más aceptada por todos los dispositivos.

NAT mode.- aquí habilitamos esta opción en el caso que debamos atravesar un NAT.
Buzón de voz.- en el caso que habilitemos este, debemos ingresar una contraseña, un correo electrónico si deseamos que el mensaje sea envía al mail.

Con todos estos campos ya llenos hacemos clic en el botón enviar y tendremos ya creada nuestra extensión lista para usar.

Configuración de tarjetas FXO.- FreePBX cuenta con una herramienta llamada **Dahdi config** está diseñado para administrar y configurar tarjetas PSTN, ya que es compatible con algunas marcas creadoras de hardware como son Digium, Rino, Openvox, Sangoma. Y trae controladores ya sea para conectarse a la PSTN en forma análoga o digital.

En nuestro caso tengo una tarjeta Digium TDM410 de cuatro puertos FXO los cuales con la herramienta nos presenta la siguiente pantalla.

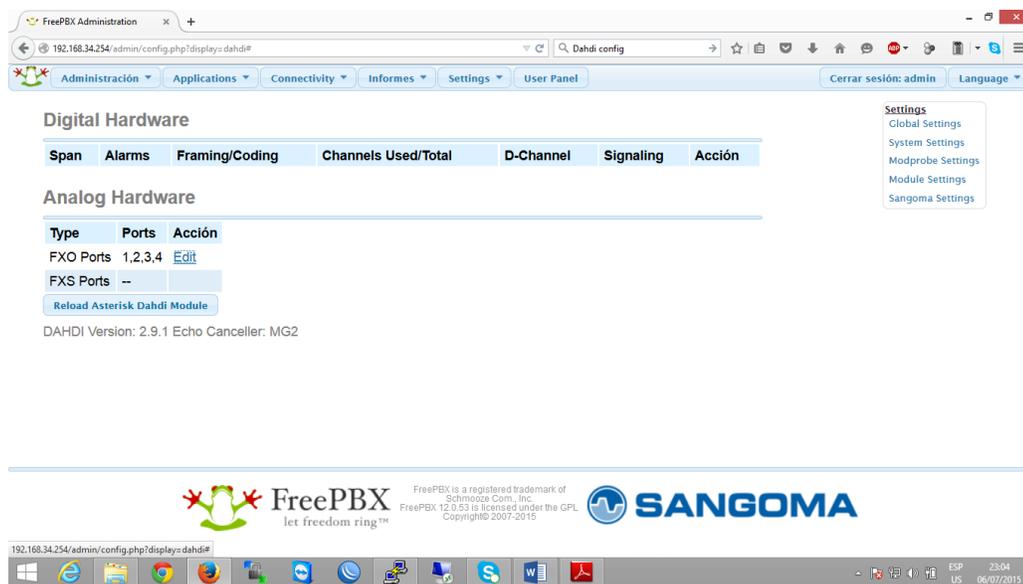


Figura 40: Configuración tarjeta análoga Fuente: Propia

Parámetros generales.- algunos de los para metros más comunes son los siguientes:

Language: idioma que se va a usar

Usercallerid: si queremos usar el identificativo de la llamada

Callwaiting: permite poner las llamadas en espera.

Transfer: permite poder transferir llamadas.

Callreturn: sirve para volver a recibir la llamada renviada sin éxito.

Echocancel: cancelación de eco activado

Echocancelwhenbridged: esta parámetro analizará cuando utilizar el canalizador de eco.

Echotrainig: tiempo en milisegundos para averiguar con cuanto de retraso viene la llamada y que cancelación de eco se debe aplicar.

Rxgain/txgain: ganancia en decibelios de la llamada entrante y saliente

Busydetect: permite detectar si el llamado está comunicado a través del tomillo BUSY.

(wikiasterisk.com, 2014)

Quedando nuestra tarjeta configurada de la siguiente manera:

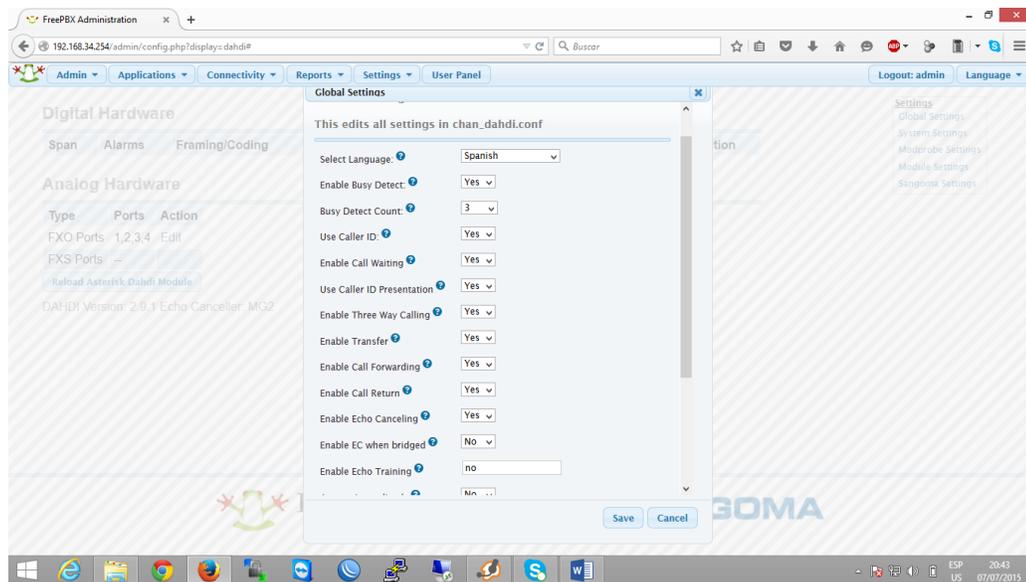


Figura 41: Configuración de tarjeta digium TDMA410 Fuente:Propia

Rutas entrantes.- nuestra PBX estará conectada al red telefónica convencional por tal motivo tendremos que saber a dónde vamos a dirigir nuestras llamadas entrantes ya sea a un IVR, una extensión, un grupo de timbrado, etc. En nuestro caso será enviada a una contestadora automática la cual nos direccionara tanto a departamentos de la institución como a realizar nuestra consulta de consumo de agua potable la cual es la finalidad de este proyecto.

Con lo cual nos dirigimos a la pestaña de conectividad, luego a rutas entrantes (Inbound routes) y nos muestra la siguiente pantalla.

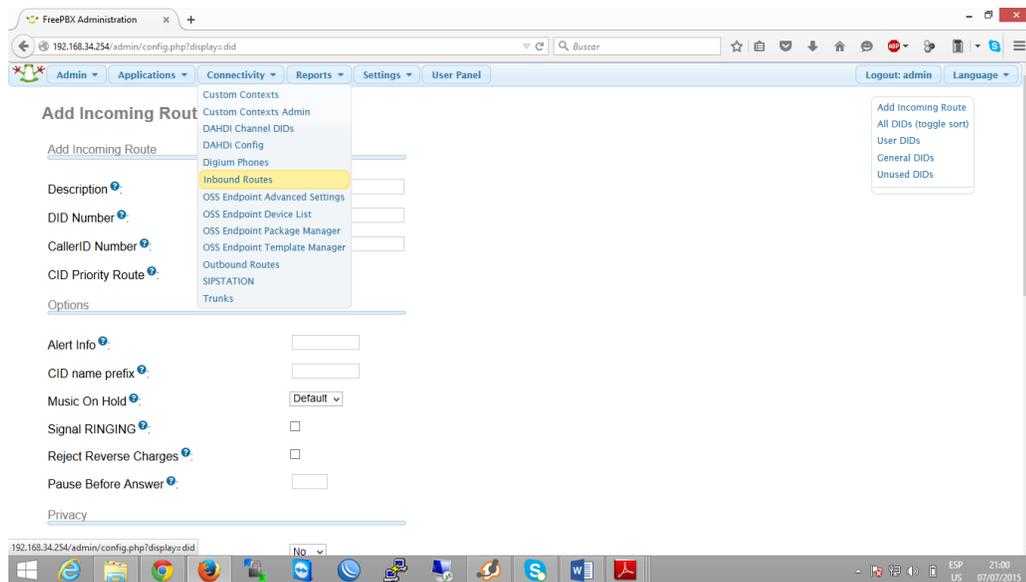


Figura 42: Ingreso a la configuración de rutas entrantes Fuente: Propia

Aquí los campos que vamos a llenar son:

Description: es un nombre o etiqueta con el cual vamos a identificar nuestras llamadas entrantes.

Set destination: aquí nos presenta algunas opciones y vamos a escoger IVR con el respectivo nombre el cual lo vamos a configurar luego.

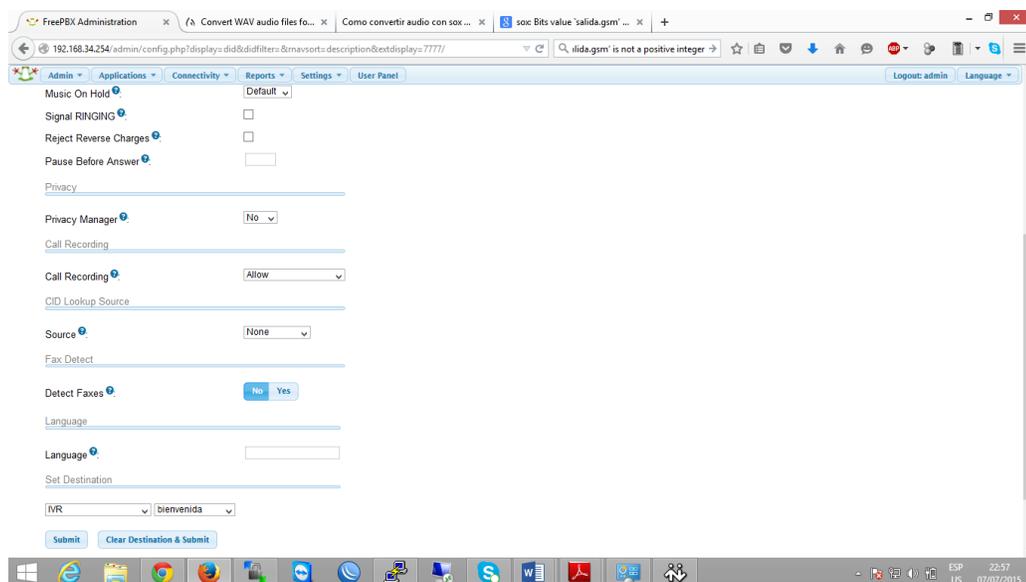


Figura 43: Llenamos los datos de nuestra ruta entrante Fuente: Propia

Con lo cual hecho esto ya tenemos direccionado el flujo de nuestras llamadas entrantes.

Rutas salientes.- para configurar las rutas salientes nos dirigimos a la pestaña de conectividad y damos clic en rutas salientes (Outbound routes).

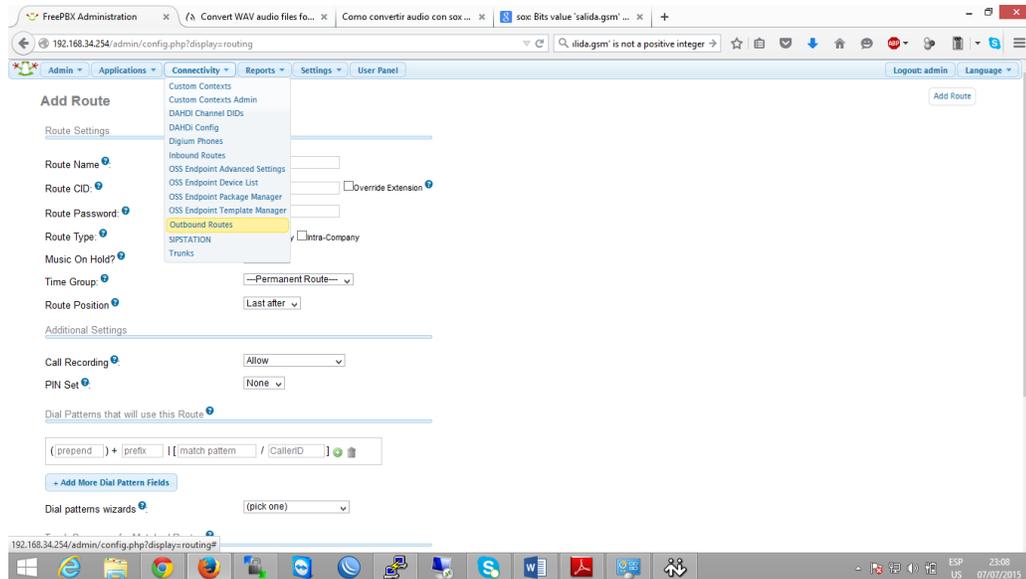


Figura 44: Pestaña para añadir ruta saliente Fuente:Propia

Los campos que vamos a llenar son:

Route name (nombre de ruta).- el cual es la etiqueta de la ruta puede ser en nuestro caso local, nacional, celular, etc.

Dial Patterns (plan de marcado).- aquí debemos tomar en cuenta que para realizar esto debemos saber el patrón para marcar a números locales, por ejemplo para nuestro caso nuestro patrón constara de 7 dígitos y tenemos nuestras reglas de marcado que es el siguiente.

X es cualquier numero entre 0-9

Z es cualquier numero entre 1-9

N es cualquier número entre 2-9

[1237-9] coincide con cualquier digito en este ejemplo 1, 2, 3, 7, 8,9

Con esto si queremos marcar 2957127 nuestro patrón será NXXXXXX Trunk (troncales).- en nuestro caso tendremos una sola troncal con el nombre g0 ya que esta que se crea por defecto cuando configuramos nuestra tarjeta FXO, ya que nuestra tarjeta tiene 4 puertos FXO crea un solo grupo y cuando hagamos una llamada tomara el puerto que esté disponible para realizar la llamada.

```

GNU nano 2.0.9 root@localhost:/var/lib/asterisk/sounds/custom
Fichero: /etc/asterisk/dahdi-channels.conf
Autogenerated by /usr/sbin/dahdi_genconf on Thu Jul 3 20:59:51 2014
; If you edit this file and execute /usr/sbin/dahdi_genconf again,
; your manual changes will be LOST.
; Dahdi Channels Configurations (chan_dahdi.conf)
;
; This is not intended to be a complete chan_dahdi.conf. Rather, it is intended
; to be #include-d by /etc/chan_dahdi.conf that will include the global settings
;

; Span 1: WCTDM/0 "Wildcard TIM410P" (MASTER)
;;; line=1 WCTDM/0/0 FXSRS (In use)"
signalling=fxs_ks
callerid=asreceived
group=0
context=from-pstn
channel -> 1
callerid=
group=
context=default

;;; line=2 WCTDM/0/1 FXSRS (In use)"
signalling=fxs_ks
callerid=asreceived
group=0
context=from-pstn
channel -> 2
callerid=
group=
context=default

;;; line=3 WCTDM/0/2 FXSRS"
signalling=fxs_ks
callerid=asreceived
group=0
context=from-pstn
channel -> 3
callerid=
group=
context=default

```

Figura 45: Archivo de configuración dahdi-channels.conf

Con estos campos llenos, nuestra ruta entrante quedaría de la siguiente manera:

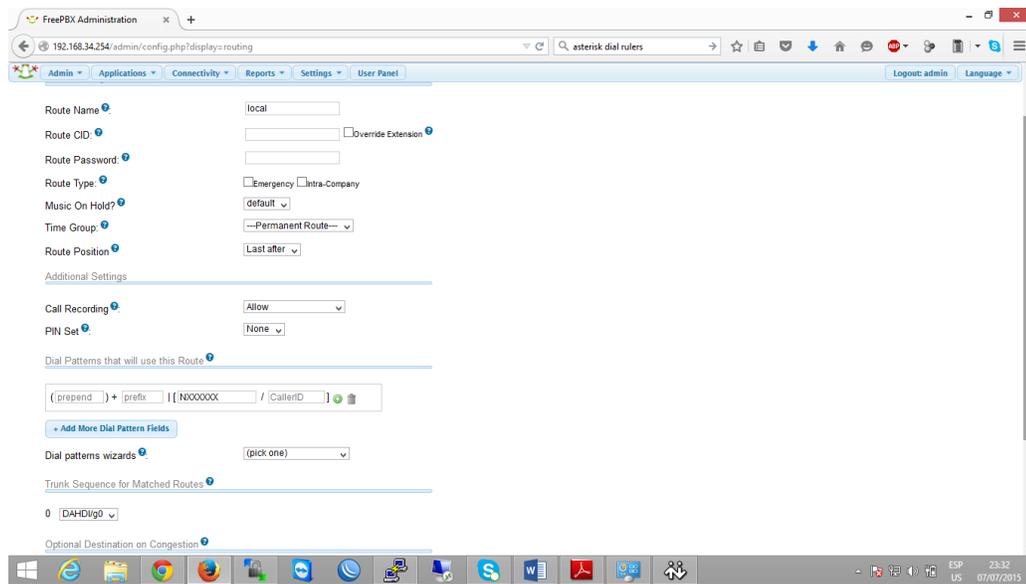


Figura 46: Configuración de ruta Saliente Fuente: Propia

Con esta configuración básica ya tendremos en funcionamiento nuestra PBX cabe recalcar que existen muchas más configuraciones que dependerá de las necesidades del cliente.

3.6 CONFIGURACIÓN DE ARCHIVOS DE ASTERISK

En este capítulo vamos a dar una explicación de los archivos principales de configuración para el funcionamiento de nuestra aplicación.

Primero debemos tomar en cuenta que nuestra conexión a la base de datos SQL Server con asterisk será por ODBC con FreeTDS el cual nos permitirá manipular dicha base de datos, además en asterisk vamos a manipular el archivo `extensions_custom.conf` el cual es el corazón del dial plan y en el cual vamos a invocar la ejecución de nuestro script que no es más un pequeño programa realizado en php el cual va a estar ubicado en la carpeta `/var/lib/asterisk/agi-bin`.

Para conectarnos con nuestra base de datos vamos a realizar el siguiente paso:

Vamos al archivo `/etc/odbcinst.ini`

En este archivo vamos a encontrar todos los drivers para las diferentes bases de datos con las que queremos trabajar por defecto vienen para postgres y mysql.

Editamos este archivo para direccionar al drivers de FreeTDS como muestra la imagen poniendo el nombre de nuestro drivers una descripción en qué lugar se encuentran los archivos. El direccionamiento dependerá si nuestro Sistema es de 32 o 64 bits

```

GNU nano 2.0.9          Fichero: odbcinst.ini
Setup                  = /usr/lib/libodbcmyS.so
Driver64               = /usr/lib64/libmyodbc5.so
Setup64                = /usr/lib64/libodbcmyS.so
FileUsage              = 1
#Driver para conectar a IVR
[FreeTDS]
Description            = FreeTDS ODBC for MSSQL
Driver                 = /usr/lib64/libtdsodbc.so
Setup                  = /usr/lib64/libtdsS.so
FileUsage              = 1

```

[^]G Ver ayuda [^]O Guardar [^]R Leer Fich [^]Y Pág Ant [^]K CortarTxt [^]C Pos actual
[^]X Salir [^]J Justificar [^]W Buscar [^]V Pág Sig [^]U PegarTxt [^]T Ortografía

Figura 47: Archivo de configuración odbcinst.ini Fuente: Propia

Ahora vamos a editar el archivo /etc/freetds.conf

En este archivo vamos a indicar el nombre con el cual vamos a etiquetar a nuestra conexión, la ip de nuestro servidor de bases de datos con la cual vamos a trabajar también el Puerto y la versión con la cual se va a conectar. Como muestra la figura

```

GNU nano 2.0.9          Fichero: freetds.conf

# A typical Sybase server
[egServer50]
    host = symachine.domain.com
    port = 5000
    tds version = 5.0

# A typical Microsoft server
[egServer70]
    host = ntmachine.domain.com
    port = 1433
    tds version = 7.0

[MSSQL]
    host = 192.168.34.2
    port = 1433
    tds version = 7.2
    client charset = UTF8

^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía

```

Figura 48: Archivo de configuración freetds.conf Fuente: Propia

Ahora vamos a editar odbc.ini

En este archivo hacemos ya nuestra conexión, indicando el nombre de nuestra conexión en corchetes, una descripción, el drivers que vamos a utilizar, el nombre de nuestro servidor, el password para conectarnos a nuestra base de datos, el Puerto, el nombre de nuestra base de datos, y el lenguaje más abajo lo podemos ver cómo quedaría editado.

```

GNU nano 2.0.9           Fichero: odbc.ini
[MSSQL]
Description = Connector ODBC
Driver = FreeTDS
Servername = MSSQL
password = tesis
port = 1433
database = tesis1
language = us_english
Trace = no

[ 9 líneas leídas ]
^G Ver ayuda  ^O Guardar   ^R Leer Fich ^Y Pág Ant   ^K CortarTxt ^C Pos actual
^X Salir     ^J Justificar ^W Buscar    ^V Pág Sig   ^U PegarTxt  ^T Ortografía

```

Figura 49: Archivo de configuración odbc.ini Fuente: Propia

Ya configurados nuestros archivos probamos la conexión a la base de datos que vamos a utilizar, digitando el siguiente comando:

```
#isql -v MSSQL tesis tesis
```

Nos mostrara la siguiente pantalla de una conexión satisfactoria

```

login as: root
root@192.168.34.254's password:
Last login: Wed May 27 20:28:10 2015 from 192.168.34.5
[root@localhost ~]# tsq1-S 192.168.34.2 -p 1433 -U tesis -P tesis
-bash: tsq1-S: no se encontró la orden
[root@localhost ~]# tsq1 -S 192.168.34.2 -p 1433 -U tesis -P tesis
locale is "es_ES.UTF-8"
locale charset is "UTF-8"
using default charset "UTF-8"
1> ^C
[root@localhost ~]# isql -v MSSQL tesis tesis
+-----+
| Connected! |
|           |
| sql-statement |
| help [tablename] |
| quit |
|           |
+-----+
SQL>

```

Figura 50: Comando para comprobar la conexión con la base de datos MSSERVER Fuente Propia

Como observamos tenemos conexión con nuestra base de datos. Ahora tenemos nuestro script hecho en php el cual se encargara de hacer la consulta para darnos resultados, este archivo lo vamos a explicar más adelante en el siguiente capítulo. Para que este script interactúe con asterisk tenemos que configurar nuestro archivo `extensions_custom.conf` el cual funciona como el Corazón de asterisk ya que ahí se realizara todo el dial plan que vamos a utilizar, vamos a crear un contexto llamado `consulta` el cual cuando marquemos la extensión 1005 nos va a responder ejecutando nuestro script `IVR.agi` y luego colgara como muestra la figura

```

GNU nano 2.0.9      Fichero: extensions_custom.conf

[from-internal]
include => from-internal-xfer
include => from-internal-custom

[from-internal-custom]
include => consulta

[consulta]
exten => 1005,1,Answer()
same => n,AGI(IVR.agi)
same => h,4,Hangup
same => i,5,Hangup

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

[internal]
include => from-internal-additional-custom
include => ext-intercom

```

[79 líneas leídas]

[^]G Ver ayuda [^]C Guardar [^]R Leer Fich [^]Y Pág Ant [^]K CortarTxt [^]C Pos actual
[^]X Salir [^]J Justificar [^]W Buscar [^]V Pág Sig [^]U PegarTxt [^]T Ortografía

Figura 51: Archivo de configuración `extensions_custom.conf` donde vamos a editar nuestro dial plan Fuente: Propia

Una vez hecho esto vamos a frontal FreePBX el cual nos facilitara la configuración de nuestro Sistema TTSIVR, buscamos la pestaña `admin` y damos clic en `custom destinations` o en español `Destinos personalizados` nos permite registrar destinos personalizados que apuntan a Dialplan y son disponibles para otros módulos. Cuando lo que vamos apuntar es a nuestro contexto “`consulta`”, extensión 1005 en la línea 1 como se muestra en la imagen. Además esta extensión ya no podrá ser utilizada por que quedara como reservada por FreePBX.

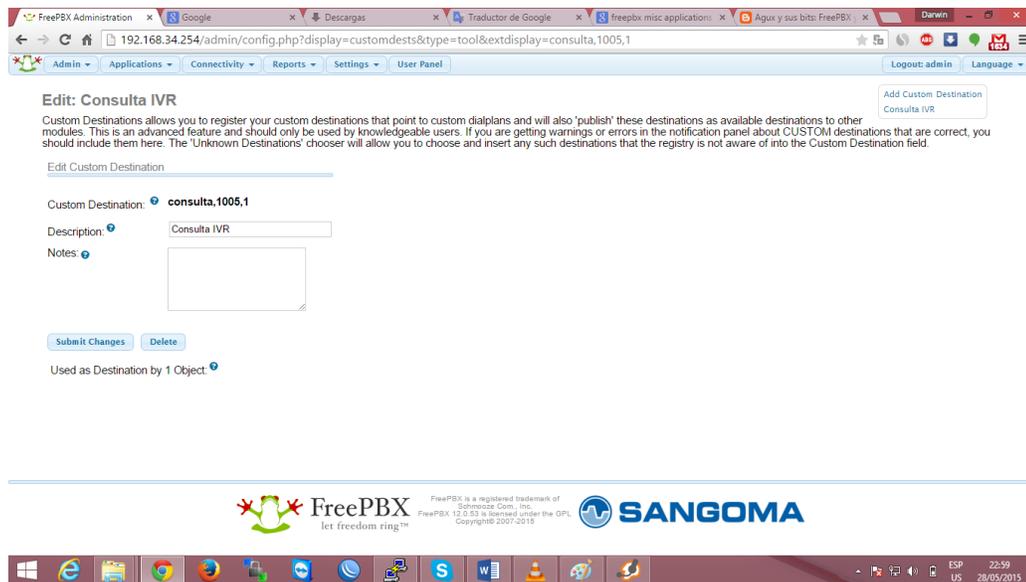


Figura 52: Configuración de Custom Destinations para nuestro IVR Fuente: Propia

Vamos a la pestaña de aplicaciones, luego a mis aplicaciones y hacemos clic en agregar

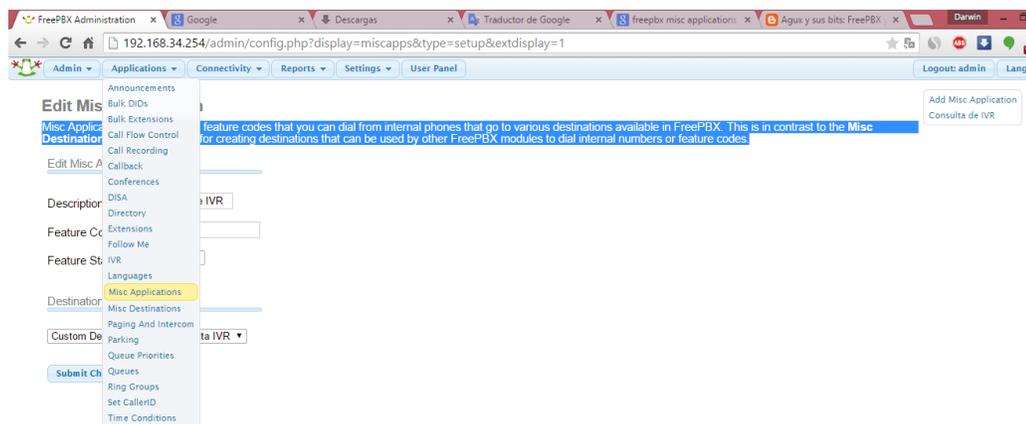


Figura 53: Descripción de cómo llegar a la pestaña de Misc Applications para nuestro IVR Fuente: Propia

Este modulo sirve para crear códigos que cuando marquemos desde cualquier teléfono internos vaya a un destino definido y pueda ser utilizado por otros módulos de FreePBX.

En la siguiente imagen muestra como quedara configurado

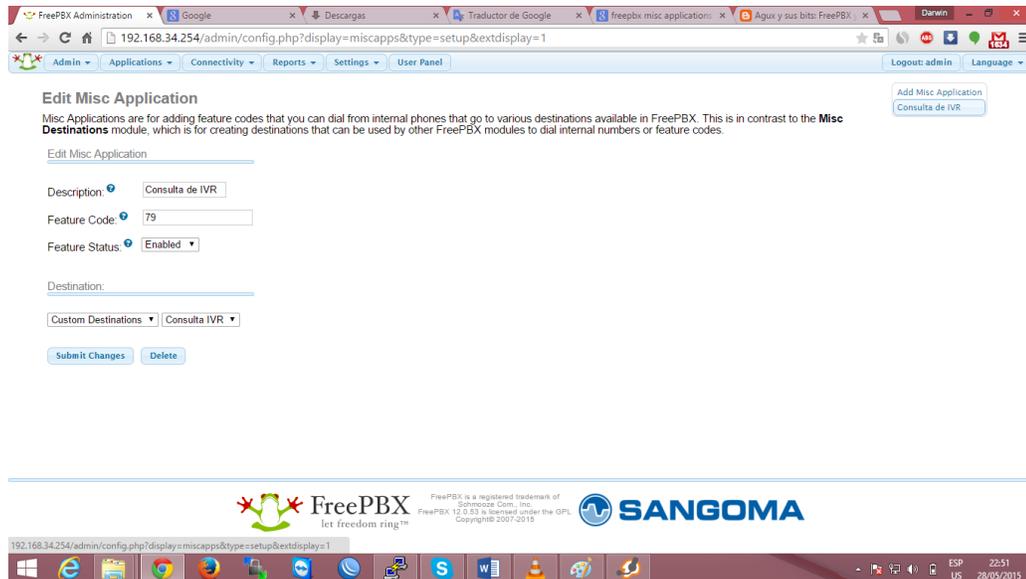


Figura 54: Configuración de Misc Applications para nuestro IVR Fuente: Propia

Es decir el momento que marquemos 79 empieza a funcionar nuestro TTS IVR
 Ahora vamos a aplicaciones, mis destinos y agregamos un nuevo y de igual manera como hicimos con “mis aplicaciones” quedara de la siguiente manera

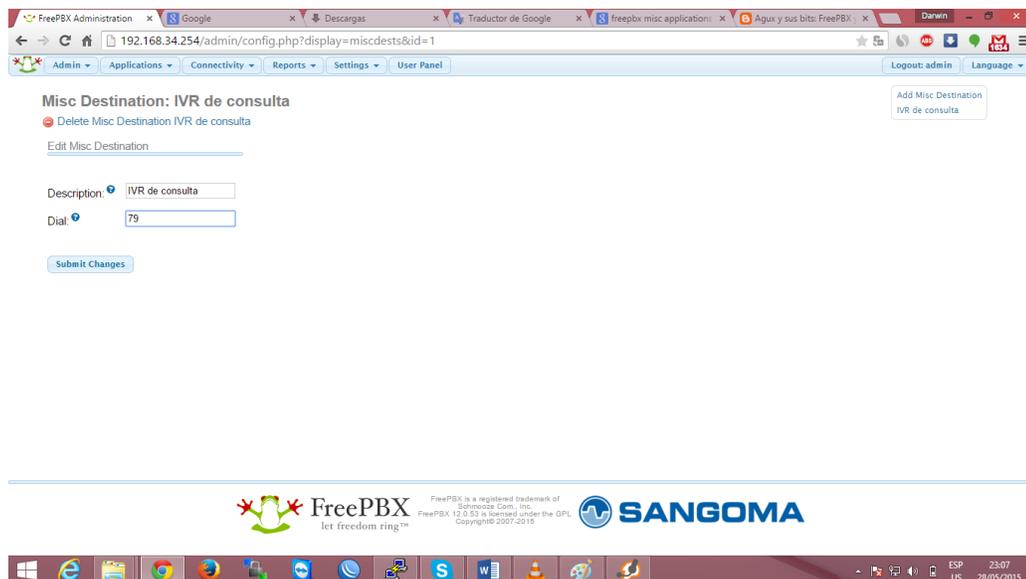


Figura 55: Configuración de Misc Destinations Fuente: Propia

Y con esto tenemos listo para empezar nuestro siguiente capítulo.

3.7 PROGRAMACIÓN DEL TTS IVR Y ENVIÓ DE MENSAJES DE TEXTO

Primero debemos tener diagramado que es lo que va a realizar nuestra aplicación en un diagrama de flujo el cual sería el siguiente:

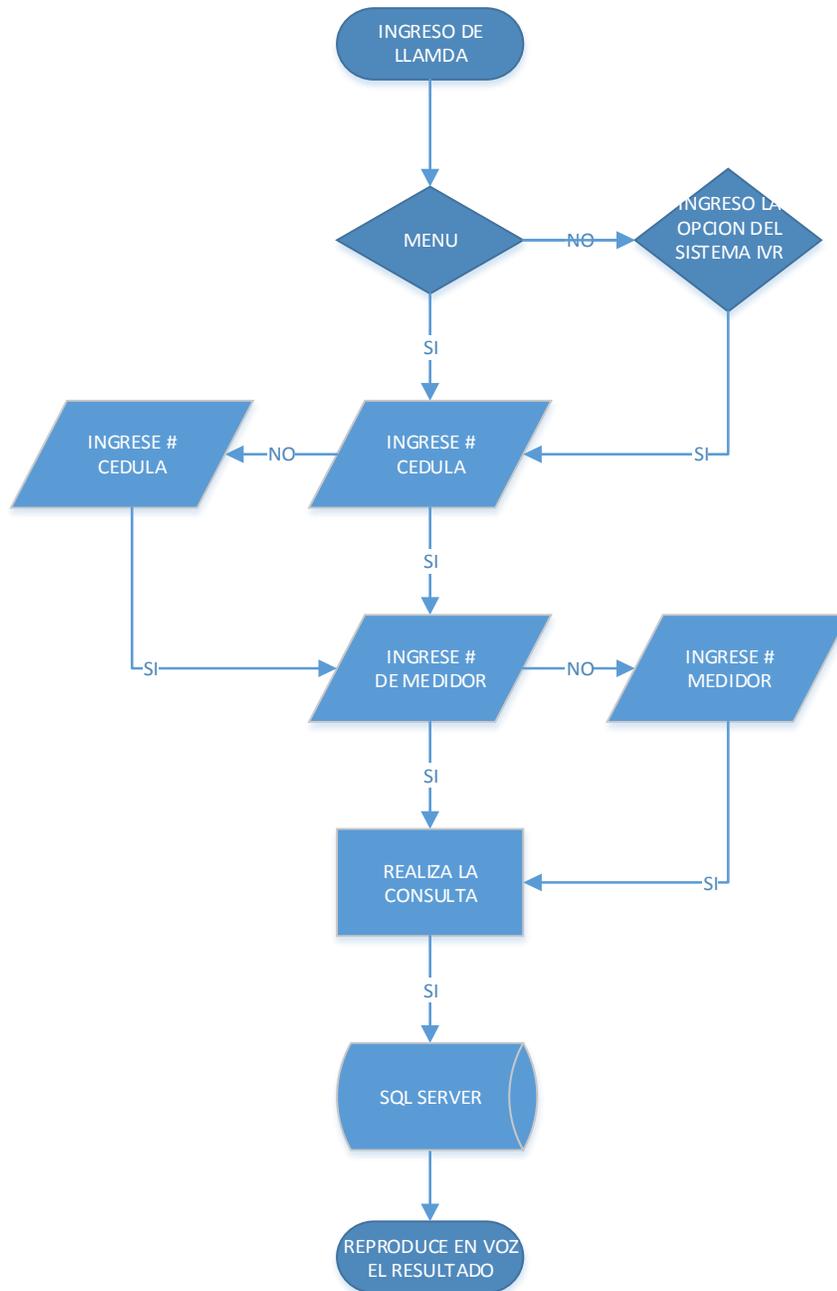


Figura 56: Diagrama de Flujo de IVR Fuente: Propia

Vamos a explicar el script el cual es capaz de transformar nuestra consulta de texto a voz:

```
#!/usr/bin/php -q
<?php
    //configuracion general
    //importar el phpagi v2.20
    require('phpagi-2.20/phpagi.php');
    set_time_limit(30);
    error_reporting(E_ALL);
    //instanciamos el agi.
    $agi = new AGI();
    $agi->answer();
    $agi->verbose(">>>>> CONSULTA: agi iniciado correctamente");
    $agi->exec_setlanguage('es');
    $agi->verbose(">>>>> CONSULTA: agi lenguaje es establecido");
    //variables de la bd
    $dbasterisk = 'tesis1';
    $servidordb = '192.168.34.2';
    $usuariodb = 'tesis';
    $passworddb = 'tesis';
    //creamos la conexion
    $agi->verbose(">>>>> CONSULTA: vars de mssql: \n db: $dbasterisk \n
    srv: $servidordb \n usr $usuariodb");
    $con = mssql_connect($servidordb,$usuariodb,$passworddb)or die(
"ERROR: NO SE PUEDE CONECTAR".mssql_get_last_message());
    //seleccionamos la base de datos
    $agi->verbose(">>>>> CONSULTA: con establecido:
    ".mssql_get_last_message()." seleccionando la bd ");
    mssql_select_db($dbasterisk,$con) or die("ERROR: NO SE
    ENCUENTRA LA BASE DE DATOS".mssql_get_last_message());
    $agi->verbose(">>>>> CONSULTA: agi iniciando saludo de bienvenida
    de grabacion");
    $agi->verbose(">>>>> CONSULTA: agi iniciando saludo de bienvenida
    de grabacion");
    //solicitar el marcado de la cedula
```

```

$correcta_cedula = 2;
while($correcta_cedula != 1){
    $sagi->exec("Background","audios-consulta/digite_cedula");
    $sagi->verbose(">>>>> CONSULTA: ingresar cedula ");
    $datos = $sagi->get_data("beep",7000,10);
    $sagi->verbose(">>>>> CONSULTA: tiempo agotado o # presionado ");
    $cedula = $datos['result'];
    //$cedula = '43904860';
    $sagi->verbose(">>>>> CONSULTA: cedula es $cedula ");
    $sagi->exec("Playback","audios-consulta/la_cedula_es");
    $sagi->exec("SayDigits",$cedula);
    $sagi->exec("Playback","audios-consulta/correcta_1_incorrecta_2");
    $datos = $sagi->get_data("beep",6000,1);
    $correcta_cedula = $datos['result'];
    // $correcta_cedula = '1';
    $sagi->verbose(">>>>> CONSULTA: el usuario marco $correcta_cedula
");
    $sagi->verbose(">>>>> CONSULTA: tiempo agotado o # presionado ");
} //fin while
//Solicitar el numero de Titulo
$correcto_titulo = 2;
while($correcto_titulo !=1){
    $sagi->exec("Background","audios-consulta/ingrese_titulo");
    $sagi->verbose(">>>>> CONSULTA: ingresar numero de titulo");
    $datos_titulo = $sagi->get_data("beep",7000,5);
    $sagi->verbose(">>>>> CONSULTA: tiempo agotado o # presionado ");
    $titulo = $datos_titulo['result'];
    $sagi->verbose(">>>>> CONSULTA: Titulo es $titulo ");
    $sagi->exec("Playback","audios-consulta/titulo_es");
    $sagi->exec("SayDigits",$titulo);
    $sagi->exec("Playback","audios-consulta/correcta_1_incorrecta_2");
    $datos_titulo = $sagi->get_data("beep",7000,1);
    $correcto_titulo = $datos_titulo['result'];
    $sagi->verbose(">>>>> CONSULTA: el usuario marco $correcto_titulo
");
}
// consultar la cedula en la bd

```



```

if($valortitulo_decimal !=0){
    $agi->exec("Playback","audios-consulta/palabra-con");
    $agi->exec("SayNumber",$valortitulo_decimal);
    $agi->exec("Playback","audios-consulta/centavos");
}

```

```
//Fecha de vencimiento de Pago
```

```

$agi->exec("Playback","audios-consulta/fecha_pago");
$agi->exec("SayUnixTime","$fechacancel,COT,ABd\digits/es_de\Y");
$agi->exec("Playback","audios-consulta/gracias_por_utilizar");
$agi->hangup();

```

```
?>
```

Para nuestro TTS IVR vamos a comenzar utilizando nuestro frontal web FreePBX vamos a aplicaciones, IVR como se muestra en la imagen

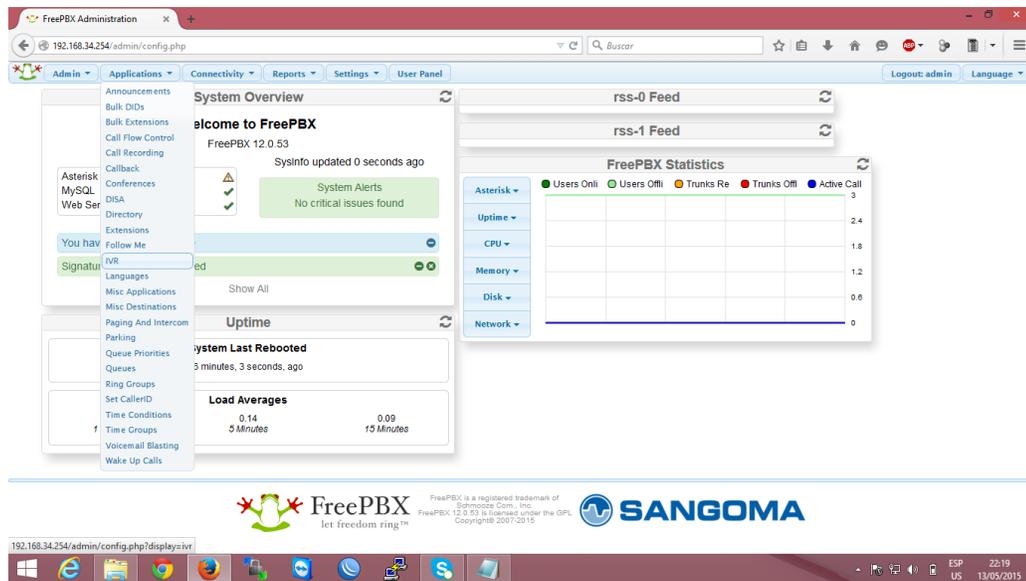


Figura 57: Descripción de cómo llegar a la pestaña de IVR para configurar nuestro IVR Fuente: Propia

Una vez ahí procedemos a añadir un nuevo IVR.

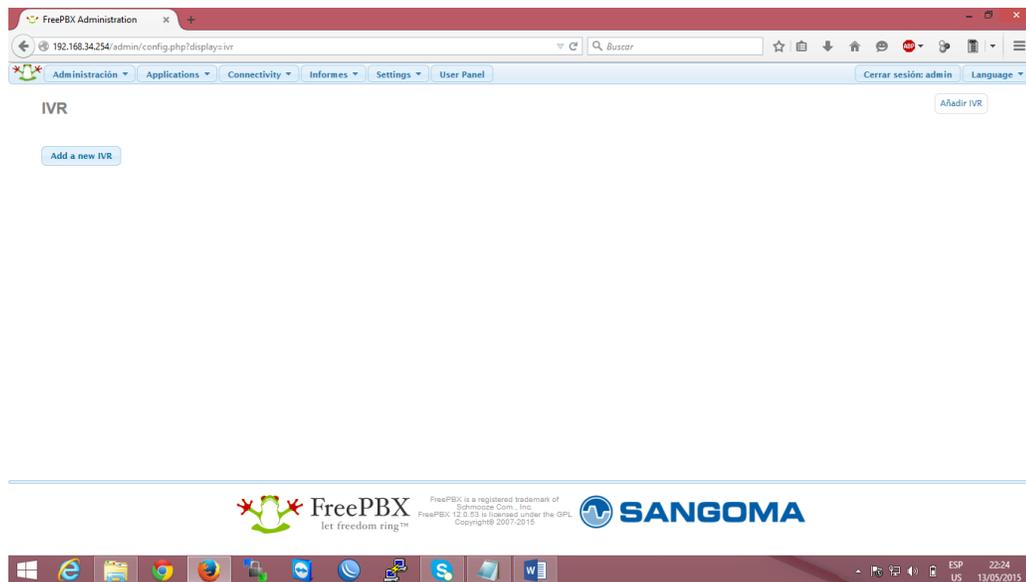


Figura 58: Pestaña de la opciones IVR para añadir nuestro primer IVR

Luego vamos a explicar y llenar los campos que se necesita para el desarrollo de nuestro TTS IVR

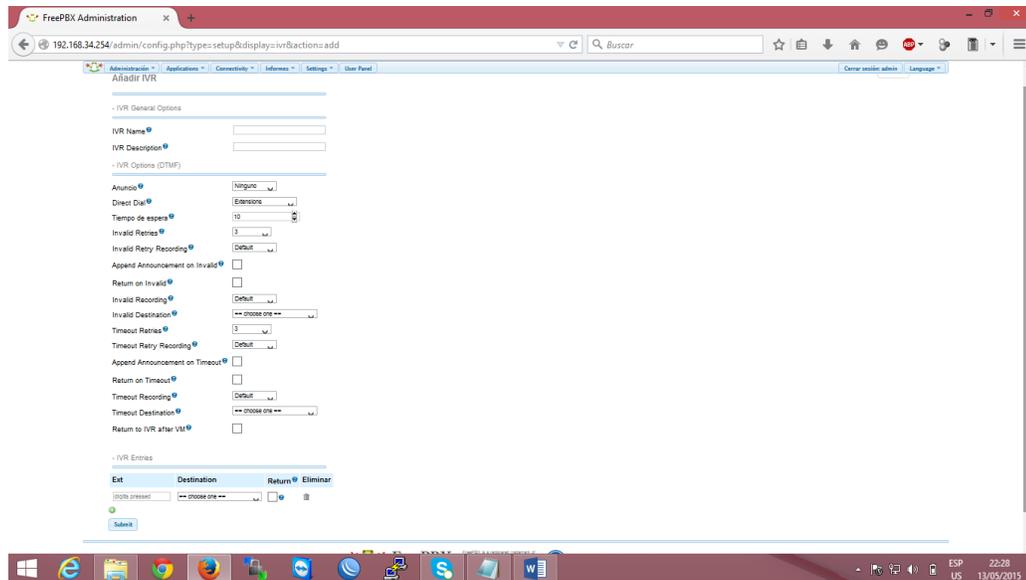


Figura 59: Nos muestra todas las opciones para la configuración de nuestro IVR

Antes de comenzar ya debemos haber ingresado nuestros anuncios de bienvenida al Sistema TTS IVR lo cual se realiza en el menú administración, grabaciones del

sistema, estos archivos deben estar en formato WAV debe estar codificado en PCM, 16 bits y a 8000Hz.

A continuación vamos a explicar los campos que vamos a llenar:

IVR Name: es el nombre con el cual vamos a definir a nuestro IVR

IVR Description: es como lo describimos a nuestro IVR

Anuncio: agréganos nuestro anuncio que hemos grabado o subido al sistema

Direct Dial: en este campo nos da varias opciones pero vamos a seleccionar extensiones para que al momento q escuchemos el anuncio podamos marcar directamente alguna extensión que conocemos.

Tiempo de espera: es el tiempo que se va a tardar en reproducir el anuncio cuando timbramos a nuestra central telefónica

Invalid Retries: número de veces para volver a intentar cuando se recibe una respuesta no válida de la persona que llama

Invalid Retry Recording: se reproduce un mensaje cuando la respuesta no es valida

Append Announcement on Invalid: reproduce nuevamente invalid retry recording

Return on Invalid: retorna nuevamente al inicio del IVR

Invalid Recording: reproduce un anuncio antes de ser enviado un destino o llega al máximo de respuestas no validas

Invalid Destinations: envía a diferentes opciones que escojamos en el menú

Timeout Retries: número de veces para volver a intentarlo cuando no se oye ningún DTMF y los tiempos de elección IVR.

Timeout Retry Recording: reproduce un anuncio que ya se acabó el tiempo máximo.

Append Announcement on Timeout:

Return on Timeout: retorna al tiempo fuera.

Timeout Recording: reproduce un anuncio de tiempo fuera

Timeout Destinations: nos presenta varias opciones cuando se termina nuestro tiempo de elección

Return to IVR after VM: nos retorna al IVR

Y por ultimo

IVR entries: son las opciones que tiene nuestro ivr cuando digitamos algún numero en particular, también debe conocer algunas extensiones especiales que maneja nuestra central para ciertos eventos.

- a: es utilizada cuando un usuario presiona '*' durante el saludo inicial del buzón de voz.
- h: cuando alguien cuelga va a esta extensión.
- i: entrada inválida
- o: extensión del operador, es utilizada en los saludos del buzón de voz
- t: cuando se agota el tiempo de selección la llamada va a esta extensión.
- T: tiempo absoluto de una llamada

Nuestras opciones serán de la siguiente manera: opción 1 que cuando marque 0 será reenviada a una operadora. Opción 2 si marca 1 se enviara a nuestro sistema de consultas. Opción 3 t seleccionamos como destino terminate call y escogemos hangup que significa colgar la llamada. Opción 4 colocamos i seleccionamos como destino terminate call y escogemos hangup que significa colgar la llamada.

3.8 ANÁLISIS DE REQUERIMIENTOS DE LA APLICACIÓN WEB DE CONSULTA DEL COMSUMO DE AGUA POTABLE.

El objetivo principal es crear una herramienta web para conocer mediante consultas, los valores que adeudan del consumo de agua potable con empresa pública de agua potable y alcantarillado del municipio de Antonio Ante.

Esta herramienta debe ser fácil e intuitiva de usar por cualquier tipo de usuario, aunque no tengan mucha familiaridad con la tecnología web.

Deberá presentar una información veraz y comprensible al usuario.

3.9 DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN

Como lo comentamos en capítulos anteriores para el aplicativo web nos vamos a adaptar a la infraestructura que poseen, que es .NET.

Como lenguaje de programación utilizaremos Visual Basic y base de datos SQL Server, los cuales son su plataforma de desarrollo y nos adaptaremos a la plantilla de su sitio web.

Nuestra aplicación contara con 3 formas de realizar la consulta de nuestro consumo de agua potable que será por clave catastral, cedula y nombre y apellido. Tal como muestra la imagen, debemos marcar por qué tipo de opción realizaremos la consulta y digitar los datos respectivamente y damos clic en consultar.

Como vemos la consulta web es muy intuitiva que cualquier usuario la podrá utilizar.

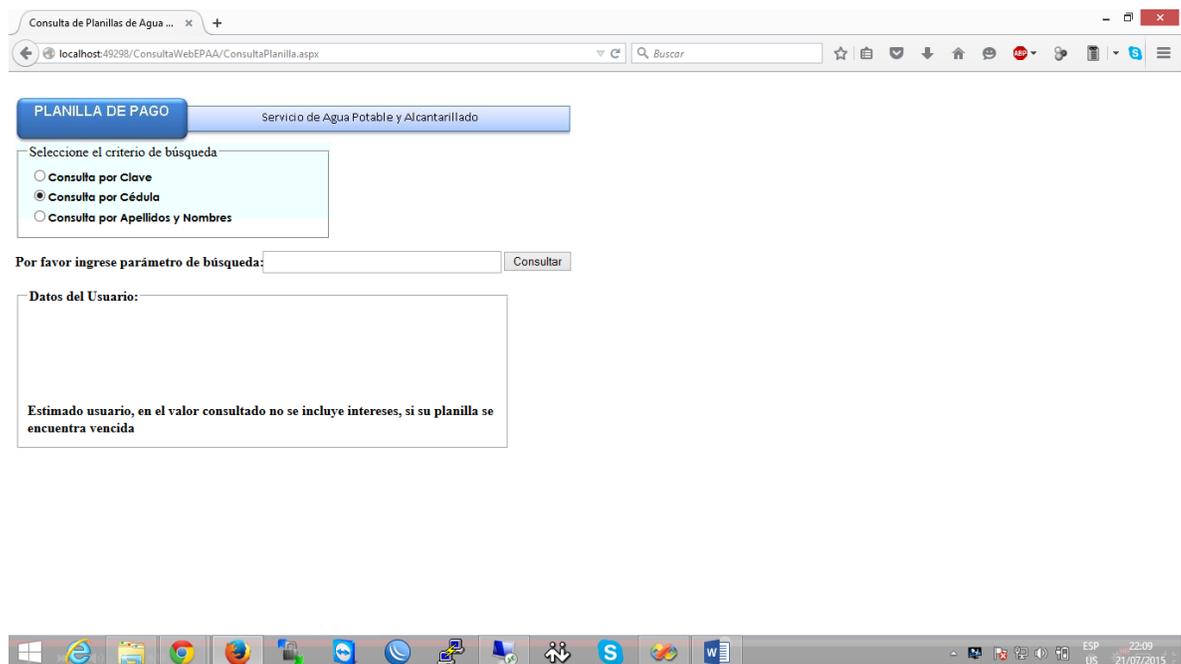


Figura 60: Pagina web de consulta de planillas de agua potable Fuente: Propia

4 CAPÍTULO: CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- El uso de la telefonía ip, en nuestro caso asterisk, nos permite interactuar con cualquier base de datos que dispongamos mediante scripts, para obtener la información de una manera fácil como es mediante una llamada telefónica, lo que con una solución privada nos costaría mucho dinero o no sería posible realizar.
- Existe un ahorro de costos para el personal administrativo, ya que no necesitamos que una persona este atendiendo la llamada las 24 horas del día.
- El uso de PHP, por ser un lenguaje de programación más usado y es el que más se adapta a Asterisk permite realizar cualquier tipo de extracción de datos de diferentes bases de datos al tener soporte estas.
- Al utilizar herramientas libres, estamos contribuyendo a establecer que las entidades públicas utilicen estas herramientas las cuales fueron decretadas como ley.
- Este tipo de proyectos mejorar la calidad de atención al cliente, ya que los usuarios no tienen que ir a las dependencias municipales para consultar el valor que adeudan, así optimizan tanto el tiempo suyo como de los empleados de dicha entidad.

4.2 RECOMENDACIONES

- Se recomienda extender el uso de las tecnologías utilizadas en este aplicativo para poder extender a otros tipos de consultas como por ejemplo: pago de patentes, impuesto predial, etc.
- Mejorar la seguridad del servidor. estableciendo políticas de seguridad como es restricción de puertos, accesos por fuerza bruta, lista de acceso para usuarios SIP/IAX, contraseñas SIP/IAX seguras mediante firewall o cualquier otro método.
- Guardar periódicamente Backups de los archivos de configuración.
- En caso de migrar las líneas telefónicas de líneas analógicas a SIP, la aplicación no se vería afectada, ya que asterisk tiene soporte para dicha tecnología y su configuración no es compleja.
- Dimensionar el hardware adecuado para el servidor, ya sea tarjetas análogas o digitales, discos duros, procesador, etc. De acuerdo al tráfico de llamadas que podrá soportar.
- La solución de mensajes de texto por el momento no es viable ya que representa un gasto al municipio y al usuario, “se recomienda implementar esta funcionalidad en el futuro”.

5 GLOSARIO DE TÉRMINOS.

AGI: Es la sigla para “Asterisk Gateway Interface”. Es el puerto del que se sirve Asterisk para conectarse con otras aplicaciones.

DID: Es la sigla para “Direct Inward Dialing”. Es el número marcado por un llamante para entrar en el sistema telefónico.

Digium, Inc.: Es una empresa privada de tecnología de comunicaciones con sede en Huntsville, Alabama. Digium especializada en el desarrollo y fabricación de hardware y software de comunicaciones de telefonía, especialmente la plataforma de código abierto de telefonía Asterisk. (Bryant, 2011)

DTMF: Multifrecuencia de doble tono. Estos tonos se usan en telefonía para marcar un número telefónico.

FXO: Es la manera de designar el punto final de conexión de la llamada. El FXO es el elemento que recibe la llamada (un teléfono o una central conectada a la PSTN). La sigla es para “Foreing eXchange Office”. La interface FXO no genera tono, solo los acepta.

FXS: Es el que envía la llamada al FXO. La sigla designa “Foreing eXchange Station”. Una interface FXS provee de tono y ringing, voltaje para alertar al equipo telefónico de una llamada entrante.

IAX: Es el protocolo o de conexión entre sistemas Asterisk.

IVR: Es la sigla para “Interactive Voice Responce”. En este manual está descrito como sistema de respuesta de recepcionista digital. Es el sistema que reproduce mensajes a los llamantes, les permite acceder a menús de voz o aplicaciones mediante la navegación por el teclado del teléfono.

NAT: Es el protocolo de traducción de direcciones de red, la sigla significa “Network Address Translation Protocol”.

PSTN: Es la sigla para “Public Switched Telephone Network”, que es el servicio de telefonía tradicional provisto por las compañías telefónicas.

SCRIPT: Es un conjunto de instrucciones, sentencias de control, variables y demás elementos de programación generalmente almacenadas en un archivo de texto (pueden considerarse como un archivo de instrucciones o como un programa).

SIP: Es la sigla para “Session Initiation Protocol”. El protocolo más popular para VoIP, utilizado por la mayoría de los productos VoIP y VSP.

SOFTPHONE: Un Softphone (en inglés combinación de Software y de Teléfono) es un software que hace una simulación de teléfono convencional por computadora. O sea permite usar la computadora para hacer llamadas a otros softphones o a otros teléfonos convencionales usando un VSP.

VoIP: Es la sigla para “Voice Over Internet Protocol”. Es lo que lleva llamadas telefónicas a través de redes de datos como redes corporativas o la Internet.

BIBLIOGRAFIA

- 3cx.es. (febrero de 2013). Obtenido de <http://www.3cx.es/voip-sip/h323/>
- asterisk.org. (mayo de 2012). Obtenido de Why IAX2?:
<https://wiki.asterisk.org/wiki/pages/viewpage.action?pageId=4817138>
- Bryant, L. M. (2011). *Asterisk Cookbook*. Gravenstein Highway North: O'Really Media.
- Cisco. (Enero de 2012). Guía del usuario de Cisco Unified Communications Manager 9.0.
 Enero.
- Correa, E. V. (2011). *Asterisk Desconsolado*. Creative Commons.
- Gomillion, D., & Barrie Dempster. (2005). *Construyendo Sistemas Telefonicos con Asterisk*. BIRMINGHAM - MUMBAI: Packt Publishing Ltd.
- Landívar, E. (2008). *Comunicaciones Unificadas con eleastix*. GNU Free Documentation License.
- msdn.microsoft.com. (Noviembre de 2007). <https://msdn.microsoft.com>. Obtenido de
<https://msdn.microsoft.com/es-es/library/4w3ex9c2%28v=vs.100%29.aspx>
- msdn.microsoft.com. (Febrero de 2015). <https://msdn.microsoft.com>.
- Perez, B. (2014). *Asterisk instalacion, configuracion y puesta en marcha*. Republica Dominicana: SNC IT.
- php.net. (Lunes de Junio de 2015). www.php.net. Obtenido de
<http://php.net/manual/es/intro-what-is.php>
- php.net. (lunes de Junio de 2015). www.php.net. Obtenido de
<http://php.net/manual/es/intro-whatcando.php>
- Russel Bryant, L. ., (2013). *Asterisk the Guia Definitive*. Unite States of America: O'Reilly.
- Simionovich, N. (2008). *Asterisk Gateway Interface 1.4 and 1.6 Programming*. Packt Publishing.
- wikiasterisk.com. (6 de julio de 2014). *DAHDI*. Obtenido de
<http://www.wikiasterisk.com/index.php/DAHDI>