

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA TELEFÓNICO
INTERACTIVO QUE PERMITA AUTOMATIZAR LA TOMA DE
PEDIDOS EN UN RESTAURANTE DE COMIDA RÁPIDA
UTILIZANDO RECONOCIMIENTO DE VOZ”**

INFORME DE MATERIA DE GRADUACIÓN

Previa a la obtención del título de:

INGENIERO EN TELEMÁTICA

Presentada por:

ALEXANDRA ELOÍSA OROZCO BUENO

ALVARO ERNESTO PADILLA VILEMA

Guayaquil - Ecuador

Año: 2011

A G R A D E C I M I E N T O

Agradezco a mis amigos y familiares por respaldarme con su apoyo incondicional y por darme ánimos siempre. Agradezco también a quienes con pequeños gestos han contribuido a que finalice con éxito esta etapa de mi vida.

ELOÍSA OROZCO B.

Agradezco a Dios por todas las bendiciones que me ha dado, a mis familiares que me han apoyado en todo momento y a mis amigos que me han ayudado a crecer como persona.

ALVARO PADILLA V.

DEDICATORIA

Dedico el esfuerzo realizado a mi mamá y especialmente a mi papá, la persona más importante para mí, quien ha sido razón y fin de cada meta trazada.

ELOÍSA OROZCO B.

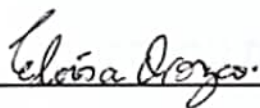
Dedico el presente a mis padres y hermanos que me han brindado su apoyo incondicional y esfuerzo. A mis profesores por compartir su conocimiento y haber sido guía en todo este camino. A mis amigos que estuvieron acompañándome y brindándome su ayuda durante tanto tiempo.

ALVARO PADILLA V.

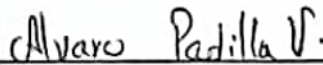
DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Trabajo de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la **Escuela Superior Politécnica del Litoral**"

(Reglamento de Graduación de la ESPOL)



Alexandra E. Orozco Bueno

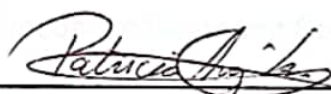


Alvaro E. Padilla Vilema

TRIBUNAL DE SUSTENTACIÓN

Ing. Gabriel Astudillo

PROFESOR DE LA MATERIA DE GRADUACIÓN



Ing. Patricia Chávez B.

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

RESUMEN

El objetivo de este proyecto fue desarrollar e implementar un sistema de respuesta de voz interactiva (IVR) que automatice el proceso de toma de pedidos en un restaurante de comida rápida utilizando reconocimiento de voz.

La solución desarrollada logró reconocer comandos de voz del usuario permitiéndole realizar registros y consultas a la base de datos y recorrer los módulos de la llamada desarrollados en Perl. Todo el software necesario utilizado fue libre: la PBX Asterisk, el reconocedor de voz Sphinx2, el motor de base de datos MySQL y el convertidor de texto a voz Festival.

INDICE DE CONTENIDO

RESUMEN _____	I
ANEXOS _____	V
ABREVIATURAS _____	VI
ÍNDICE DE FIGURAS _____	VII
INTRODUCCIÓN _____	X
CAPÍTULO 1	
ANÁLISIS CONTEXTUAL _____	1
1.1. Antecedentes y Justificación. _____	1
1.2. Objetivos del proyecto _____	5
1.2.1. Objetivo General _____	5
1.2.2. Objetivos específicos _____	5
1.3. Metodología _____	6
1.4. Perfil de la tesis _____	7
CAPÍTULO 2	
2. MARCO TEÓRICO _____	9
2.1. Asterisk _____	9
2.1.1. Características de llamada disponibles _____	9
2.1.2. Plan de mercado _____	11
2.1.3. Arquitectura de Asterisk. _____	12
2.1.4. AGI _____	13
2.2. VoiceXML _____	16
2.2.1. Objetivos del VoiceXML. _____	16
2.2.2. Aspectos de una aplicación VoiceXML. _____	17
2.2.3. Gramáticas de VXML. _____	17
2.3. Base de Datos Mysql. _____	19
2.4. Software de Reconocimiento de Voz _____	20
2.4.1. Sphinx _____	20
2.4.2. Modelo Acústico. _____	20
2.4.3. Modelo de Lenguaje _____	21
2.5. Comprensión del Lenguaje Natural – NLU _____	22
2.6. Festival _____	22
2.7. Perl _____	23

2.8.	PHP	23
2.9.	Smarty	24
2.10.	Softphone	24
CAPÍTULO 3		
3.	ESPECIFICACIONES Y HERRAMIENTAS PARA LA SOLUCIÓN.	25
3.1.	Esquema General de la Solución	25
3.2.	Especificaciones del servidor	27
3.3.	Modelo de la base de datos	28
3.4.	Instalación de Asterisk	34
3.5.	Componentes necesarios para la aplicación web	36
3.5.1	Instalación de PHP	36
3.6.	Vxml	37
3.7.	Sphinx2	38
3.8.	Esquemas de los scripts AGI	43
3.8.1.	Opciones del menú principal	43
3.8.2.	Bienvenida	45
3.8.3.	Autenticar Cliente	46
3.8.4.	Menú Principal	48
3.8.5.	Menú de Categorías	48
3.8.6.	Menú de ítems	52
3.8.7.	Listar ítems	52
3.8.8.	Presentar ítems para mostrar	54
3.8.9.	Confirmar Pedido	57
3.8.10.	Revisar estado de pedidos pendientes	58
3.8.11.	Revisar detalles de un pedido	61
3.8.12.	Más información	63
3.8.13.	Chao: Cerrar llamada	63
3.8.14.	Error	66
CAPÍTULO 4		
4.	PRUEBAS Y RESULTADOS	68
4.1.	Registro de un usuario	68
4.2.	Selección de un pedido	72
4.3.	Revisión de un pedido	77
4.4.	Inicio de sesión en la aplicación interna de empleados	79

4.5.	Menú principal de la aplicación web	80
4.6.	Reporte principal: Estados de pedidos pendientes	81
4.7.	Detalles del pedido seleccionado	82
4.8.	Listado de clientes existentes	83

CONCLUSIONES Y RECOMENDACIONES

GLOSARIO

REFERENCIAS BIBLIOGRÁFICAS

ANEXOS

1. **Anexo 1:** ESPECIFICACIONES DEL SERVIDOR
2. **Anexo 2:** ELEMENTOS DE RECONOCIMIENTO DE VOZ
3. **Anexo 3:** CODIGO DE LA BASE DE DATOS
4. **Anexo 4:** EXTENSIONS.CONF
5. **Anexo 5:** SCRIPTS AGI
6. **Anexo 6:** CODIGO DE LA APLICACIÓN WEB INTERNA

ABREVIATURAS

AGI: Asterisk Gateway Interface / Interfaz de puerta de enlace de Asterisk

AM: Acoustic Model / Modelo acústico

ASR: Automatic Speech Recognition / Reconocimiento automático de voz

CENTOS: Community ENTerprise Operating System / Sistema Operativo de comunidad empresarial

CDR: Call Detail Records / Registros de detalle de llamadas

DTMF: Dual-tone multi-frequency / Multi frecuencia doble tono

GPL: GNU General Public License / Licencia pública general del proyecto GNU

IVR: Interactive Voice Response / Respuesta de voz interactiva

LM: Language Model / Modelo de lenguaje

PBX: Private branch exchange / Ramal privado de conmutación automática

PERL: Practical Extraction and Report Language / Lenguaje Práctico para la Extracción e Informe

PHP: Hypertext Pre-processor / Preprocesador de hipertexto

SIP: Session Initiation Protocol / Protocolo de Inicio de Sesión

SRE: Speech Recognition Engine / Motor de reconocimiento del habla

STT: Speech To Text / Voz a Texto

TTS: Text To Speech / Texto a Voz

VoIP: Voice over IP / Voz sobre IP

ÍNDICE DE FIGURAS

Figura 2.1: Interacción de diferentes contextos.....	12
Figura 2.2: Representación del intercambio de datos entre Asterisk y AGI.....	14
Figura 2.3: Representación del intercambio de datos entre Asterisk y AGI.....	16
Figura 2.4: Ejemplo de reconocimiento de voz usando VXML.....	19
Figura 3.1: Esquema general de la solución.....	26
Figura 3.2: Modelo físico de la base de datos.....	29
Figura 3.3: Webservice Lmtool.....	41
Figura 3.4: Menú principal.....	44
Figura 3.5: Bienvenida.....	45
Figura 3.6 Autenticar Cliente.....	47
Figura 3.7: Menú principal.....	49
Figura 3.8: Menú categorías.....	51
Figura 3.9: Menú de Items.....	53
Figura 3.10: Listar Items.....	54
Figura 3.11: Presentar ítems para seleccionar.....	56
Figura 3.13 Confirmar Pedido.....	59
Figura 3.14 Revisar estado de Pedido Pendiente.....	60
Figura 3.15 Oír detalles de un pedido.....	62

Figura 3.16 Más información.....	64
Figura 3.17 Chao: Cerrar llamada.....	65
Figura 3.18 Gráfico de Error.....	67
Figura 4.1 Llamada al sistema desde el softphone.....	68
Figura 4.2 CLI de Asterisk: Recepción de la llamada.....	69
Figura 4.3 Ingreso del número de cédula en el softphone.....	69
Figura 4.4 CLI de Asterisk: Lectura de número de cédula.....	70
Figura 4.5 CLI de Asterisk: Ingreso y confirmación de contraseña.....	71
Figura 4.6 Ingreso y confirmación de contraseña desde el softphone.....	71
Figura 4.7 CLI de Asterisk: Selección en menú principal	72
Figura 4.8 CLI de Asterisk: Menú de Categorías.....	73
Figura 4.9 CLI de Asterisk: Selección de ítems.....	73
Figura 4.10 CLI de Asterisk: Selección de ítems. Selecciona confirmar pedido.....	74
Figura 4.11 CLI de Asterisk: Confirmar pedido.....	75
Figura 4.12 CLI de Asterisk: Guardar pedido.....	76
Figura 4.13 CLI de Asterisk: Se registró el pedido, va a menú principal.....	76
Figura 4.14 CLI de Asterisk: Revisar estado de pedidos pendientes.....	77
Figura 4.15 CLI de Asterisk: Revisar detalles de pedido seleccionado.....	77
Figura 4.16 CLI de Asterisk: Listado de detalles del pedido.....	78
Figura 4.17 CLI de Asterisk: Se cierra la llamada.....	78

Figura 4.18 Pantalla de Inicio de Sesión de aplicación interna.....	79
Figura 4.19 Pantalla principal de aplicación interna.....	80
Figura 4.20 Actualizar estado de pedidos pendientes.....	81
Figura 4.21 Detalles del Pedido.....	82
Figura 4.22 Listado de clientes.....	83

INTRODUCCIÓN

Un objetivo primordial en el que invierten las empresas de mediano y gran tamaño es posicionarse frente a la competencia optimizando la comunicación con los clientes. A partir del boom de las punto com se creó la necesidad de tener una presencia virtual definida y bien posicionada. Para lograr este objetivo se utilizan estrategias como forjar una identidad de marca y crear un vínculo fuerte con los clientes. Además, se busca multiplicar los canales de comunicación directos unilaterales y bilaterales con los clientes. Se utilizan los unilaterales para brindar información para darse a conocer, difundir actualizaciones (ya sean anuncios, ofertas o premios) o para recibir información como la retroalimentación del cliente y los bilaterales para brindar soporte al usuario.

Los medios tradicionales para comunicarse con los clientes van desde el correo físico, las centrales telefónicas, y el correo electrónico hasta la radio y televisión. Actualmente las empresas fortalecen su presencia en internet ofreciendo información o servicios en sus páginas web y también en las múltiples redes sociales existentes. Aunque todos estos medios son efectivos para hacer llegar información al cliente, la vía telefónica es el medio preferido por los usuarios

cuando buscan recibir soporte de manera inmediata y personalizada.

Por este motivo negocios de cualquier índole que cuenten con un volumen de clientes realizando llamadas telefónicas mayor al número de operadores disponibles para contestarlas consideran una opción conveniente implementar un sistema de contestación automática.

Este sistema interactivo permite a las organizaciones optimizar recursos porque al automatizar un proceso rutinario ayuda a que cada empleado se enfoque más en las actividades prioritarias de su cargo. De esta manera se aumenta la eficiencia de los procesos de la organización, en este caso un restaurante de comida rápida al que llamaremos McVoip.

CAPÍTULO 1

ANÁLISIS CONTEXTUAL

1.1. Antecedentes y Justificación.

En nuestro medio, la toma de pedidos en restaurantes de comida rápida, en especial de pizzerías, se realiza comúnmente de manera presencial, a través de un portal web o por vía telefónica. En el caso de la llamada telefónica, el empleado del restaurante debe pedir los datos completos de la persona que realiza la llamada si es un cliente nuevo o buscarlo en una aplicación si éste ya se encuentra registrado. Después de que el pedido sea definido y se haya especificado la forma de pago, éste debe ser despachado para la entrega.

La aplicación de despacho de pedidos debe tener una interfaz amigable y de uso fácil y rápido para sus usuarios - los empleados del restaurante. Al ser un restaurante de comida rápida se dará el escenario donde pocos empleados deberán encargarse del manejo de varios pedidos en poco tiempo.

Una estrategia actual para maximizar la satisfacción del cliente es ofrecerle no sólo un producto, sino un beneficio. Este beneficio comúnmente es el servicio que rodea al producto. En las cadenas de restaurantes de comida rápida, los

productos clave son alimentos sencillos de fácil preparación que en el caso de muchos restaurantes ya están precocidos y listos para calentar.

El servicio en el restaurante consiste en minimizar los tiempos de preparación y entrega, atendiendo siempre al cliente con una sonrisa. En el caso del pedido telefónico para entregas a domicilio, el servicio se basa en satisfacer todos los requerimientos que pueda solicitar un cliente sin que sienta la necesidad de presentarse físicamente en el restaurante para lograr su objetivo. Al automatizar la recepción de llamadas entonces, se debe mantener la amigabilidad que caracteriza al restaurante tratando de personalizar lo más posible la llamada manteniendo a la vez un menú intuitivo y directo.

Jacques Horovitz, experto en estrategias de servicio al cliente, resalta la diferencia de amigabilidad y sencillez de un mismo sistema con este ejemplo [1]: "American Airlines decidió reducir costos de distribución. La compañía inventó una máquina que le daba el pasaje al cliente, sin necesidad de que éste tuviera que ir a la agencia de viajes. Pasé por una de esas máquinas e introduje mi tarjeta de crédito. El aparato me dijo: 'Hola señor Horovitz' y me preguntó a dónde quería volar. Al tipear las tres primeras letras me dio todos los aeropuertos del mundo que empiezan así, para que no se me cansaran los dedos. Luego preguntó en qué clase y sector prefería el pasaje, y me dio mi pasaje 60 días antes de la partida. Como ven, la máquina me dijo hola, evitó que

me cansara y me ayudó a conseguir mi pasaje sin ningún obstáculo.

Cuatro días después, en un aeropuerto de Francia, fui a la máquina Elise de Air France para probarla. Inserté mi tarjeta de crédito y en la pantalla decía sólo VISA; no me dijo 'Hola', no me ayudó a tipear, y cuando terminé de escribir, me preguntó cuál era mi código secreto para la reserva. En ese momento, saqué la tarjeta antes de que me la tragara y leí que en las instrucciones decía: "Tiene que llamar primero a Air France. Si contestan el teléfono antes de que suene veintisiete veces, usted podrá hacer su reserva telefónicamente y le darán un número de código secreto de doce dígitos que usted tiene que anotar y registrar en la máquina del aeropuerto dos horas antes de su partida. De lo contrario su reserva se cancela."

En este proyecto, la personalización consiste en la atención a algunos detalles, por ejemplo, si un usuario se ha autenticado, se debería poder revisar su género en la base de datos, para poder decirle 'Bienvenido' o 'Bienvenida ' junto a su nombre. Si existe solo un pedido pendiente, se dice: 'Estado del pedido pendiente' en lugar de 'Estado de los pedidos pendientes'. El nivel de detalle de la personalización puede alargar el código pero es preferible una conversación muy detallada a una muy genérica.

El nivel de complejidad de la aplicación dependerá de varios factores que

podrían ser requeridos, como la necesidad de verificar el pedido, la opción de crear diferentes tipos de usuarios, especificación de la forma de pago o generación de reportes, por nombrar algunos.

El alcance de esta solución se detalla al explicar la lógica del negocio en el capítulo 3. Vale recalcar que no es nuestro objetivo enfocarnos en la generación de reportes detallados a excepción del de los pedidos pendientes pero el diseño de este sistema se basa en una interpretación flexible de la lógica de negocio que puede permitir la elaboración de reportes variados.

En este proyecto se busca unificar procesos que utilizan diferentes tecnologías con el fin de recorrer las distintas opciones de la llamada e interactuar con la base de datos bajo el control del usuario por comandos de voz.

1.2. Objetivos del proyecto

1.2.1. Objetivo General

Diseñar e implementar una aplicación que automatice el proceso de toma de pedidos en un restaurante permitiendo a los clientes interactuar con el sistema con comandos de voz.

1.2.2. Objetivos específicos

Para alcanzar nuestro objetivo general se debe:

- Diseñar un modelo de toma de pedidos con la lógica de negocio de un restaurante de comida rápida.
- Implementar un asistente automático de llamadas con respuesta de voz (IVR) que presente menús y submenús por medio del ingreso y extracción limitados de datos de la base del restaurante.
- Integrar reconocimiento de voz a nuestro sistema para permitir al usuario seleccionar los menús y opciones con comandos de voz.
- Mantener siempre una navegación fluida e intuitiva para el usuario.
- Implementar una aplicación interna para uso de los empleados del restaurante que muestre un reporte de estado de los pedidos

pendientes y permita modificar el estado de los mismos.

1.3. Metodología

Para desarrollar una aplicación que automatice un proceso existente se debe estudiar el comportamiento de ese proceso actualmente, identificando qué requerimientos satisface y cómo lo logra. Al conocer las rutinas que el usuario está acostumbrado a oír, se debe reconocer cuáles son los requerimientos esenciales y los deseables para simular este proceso y optimizarlo.

De un restaurante de comidas rápidas se puede asumir que las necesidades esenciales son revisar los productos disponibles y sus precios para luego seleccionar los que se comprará. Una vez armado el pedido sería deseable que el cliente tenga la oportunidad de confirmar si lo llevará finalmente o no. Una necesidad es que el cliente tenga acceso al listado de pedidos que ha creado para comprobar si un pedido se ha guardado correctamente.

Al identificar a todos los elementos que intervienen para lograr el éxito de la llamada, se debe ir modelando la base de datos. A la vez se debe esquematizar la opciones que tendrá el cliente y bajo qué restricciones. Por ejemplo, que sólo

los usuarios registrados tengan la opción de comprar pedidos. De esta manera se define los elementos de la base de datos, y se construye la estructura de menús y opciones.

La base se puede expandir a medida que se agreguen más funcionalidades. Las opciones de un menú se comienzan a implementar como extensiones del dialplan para probarlas y luego se modularizan en scripts AGI que naveguen fluidamente entre ellos.

Se implementará esta solución en un servidor con el sistema operativo libre Centos, el PBX Asterisk, y el gestor de base de datos MySQL. Se utilizará el motor de reconocimiento de voz Sphinx2 por su capacidad de integrarse con Asterisk.

1.4. Perfil de la tesis

El fin de esta tesis es automatizar un proceso de toma de pedidos optimizando recursos a empleados y clientes de un restaurante.

Esto lo intentamos desligando a los empleados del restaurante del proceso de presentar opciones explicando disponibilidad y precios de distintos ítems, anotar los detalles del pedido, y hacer rectificaciones al mismo antes de confirmarlo. Este proceso se vuelve monótono en el caso de que el empleado del

restaurante deba realizar el mismo proceso sin variaciones con cada uno de los clientes vía telefónica.

Al cliente se le está dando la facilidad de realizar y revisar cuantas veces quiera un pedido de forma automatizada desde la comodidad de su hogar. En el caso de que el cliente quiera revisar el menú del restaurante pero no necesariamente vaya a realizar una compra, para el cliente esto es menos molesto que preguntarle a un operador humano.

Para el restaurante como empresa, la mejora es ofrecer un servicio más a sus clientes, en este caso uno innovativo que actualmente la competencia no brinda. La principal optimización sería la asignación de recursos; los empleados ahora deberían tener como tareas únicas la atención física en el restaurante y la elaboración y entrega de pedidos. Sólo en casos excepcionales, como en errores de elección y confirmación por parte del usuario o algún fallo en el acceso al servidor se atenderían personalmente las llamadas.

La solución se implementará unificando el reconocimiento de voz para recibir comandos del usuario que sirvan para el procesamiento de la llamada y el intercambio de datos limitado con la base de datos.

CAPÍTULO 2

2. MARCO TEÓRICO

2.1. Asterisk

Asterisk es una implementación en software de una PBX multiplataforma pero originalmente pensada para GNU/LINUX.

Asterisk tiene funcionalidades basadas en los principales servicios que presta: Telefonía IP, Servidor de Correo, Servidor de Fax, Conferencias, Servidor de Mensajería Instantánea

Este software es una solución viable para compañías que necesitan tener una red telefónica privada separada de la red de telefonía local. De esta manera se ahorra el empleo de una línea propia con salidas a la red pública, reduciendo costos considerables dependiendo del tamaño de la empresa.

2.1.1. Características de llamada disponibles

Asterisk ofrece funcionalidades básicas de una PBX y características

más avanzadas así como la interoperabilidad con sistemas básicos de telefonía común y sistemas VoIP, como se hace referencia en [3].

A continuación presentamos un breve listado de las tantas características que ofrece Asterisk.

- Correo de voz
- Conferencias
- Llamadas en espera
- Grabado de llamadas
- Respuesta Interactiva de Voz
- Marcado por nombre
- Integración de base de datos
- Mensajes SMS (Short Message Service)
- Creación y transferencia de música
- Recepción de llamadas lejanas
- Fax transmitidos y recibidos
- Lista de directorios interactivos
- Privacidad

2.1.2. Plan de mercado

El plan de mercado o mejor conocido como dialplan es donde reside toda la lógica de nuestra central telefónica.

Todas las llamadas realizadas a través de Asterisk son manejadas por circuitos lógicos de voz, que se los conoce como canales. El propósito de Asterisk es manejar el tráfico de voz de acuerdo a un conjunto de reglas, como se hace referencia en [4]. El plan de mercado controla toda la secuencia que debe de seguir una llamada desde su inicio hasta su finalización. El dialplan de Asterisk incluyen reglas específicas de qué hacer cuando:

- Una llamada se recibe de un usuario en particular.
- Si una llamada se realiza a una determinada fecha y hora.
- El extremo receptor no contesta durante un intervalo de tiempo.
- Se presenta al usuario cierto menú, y este tiene que presionar ciertas teclas del teléfono.
- Y muchas situaciones más.

Todo el plan de mercado se encuentra descrito en el archivo `extensions.conf` y consta de 3 secciones:

- `[general]`
- `[globals]`

➤ [mi-contexto]

En el contexto general definiremos parámetros que se aplicarán en todos los contextos posteriores. En el contexto globales se definen todas las variables globales que se utilizarán en el plan de marcado.

Se pueden crear contextos a nuestra conveniencia para ejecutar varias operaciones en el plan de marcado. Las extensiones pertenecen a un contexto, y estos pueden incluirse en otros contextos o saltar a otro contexto como podemos visualizar en la figura 2.1.

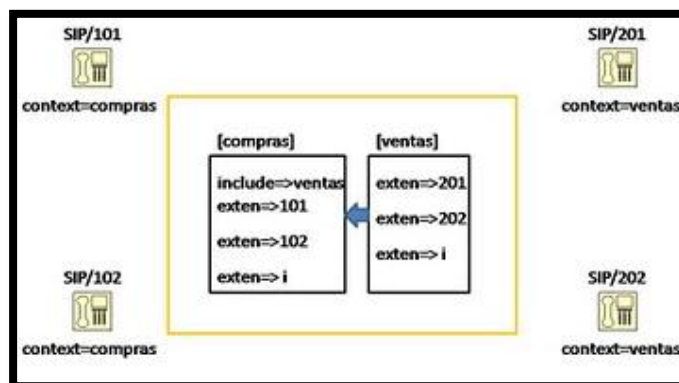


Figura 2.1: Interacción de diferentes contextos en Asterisk [5].

2.1.3. Arquitectura de Asterisk.

Asterisk cuenta con varios directorios donde se encuentran diversos archivos necesarios para la configuración de diferentes servicios, como se hace referencia en [6].

Entre los más importantes para nuestra aplicación tenemos 2 directorios:

➤ ***/etc/asterisk/***

En este directorio podemos encontrar todos los archivos de configuración de nuestra central telefónica (extensions.conf, sip.conf, entre otros).

➤ ***/var/lib/asterisk/agi-bin/***

Todos los scripts de AGI a utilizarse en el sistema se encuentran en este directorio.

2.1.4. AGI

El AGI (Interfaz de puerta de enlace de Asterisk) provee una interfaz estándar para poder enlazar aplicaciones externas con el plan de marcado de Asterisk. Generalmente, los scripts AGI se utilizan para realizar lógica avanzada, comunicarse con base de datos, etc.

Los lenguajes más comunes de programación de scripts AGI son: PHP, Python y Perl, aunque se puede utilizar cualquier otro lenguaje, como se hace referencia en [7].

El intercambio de información del script con Asterisk se realiza vía los

canales de comunicación: STDIN, STDOUT y STDERR.

- Lee desde STDIN para obtener información.
- Escribe en STDOUT para enviar información.
- Escribe en STDERR para enviar información de depuración.

El script AGI envía comandos a Asterisk escribiendo en el STDOUT. Seguidamente Asterisk envía una respuesta por cada uno de ellos que es leída por el script, como se puede visualizar en la figura 2.2.

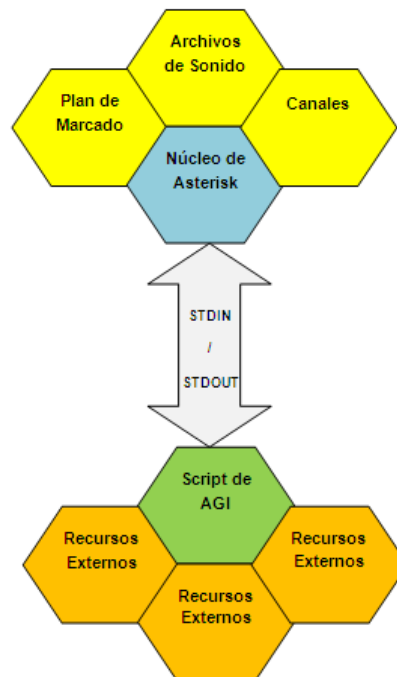


Figura 2.2: Representación del intercambio de datos entre Asterisk y AGI.

Entre algunos comandos tenemos:

- ANSWER: atiende.
- HANGUP: cuelga.
- SAY [NUMBER | DIGITS | ALPHA | PHONETICS]: dice un número, dígito, carácter o una cadena fonéticamente.
- SET [CONTEXT | EXTENSION | PRIORITY]: establece un nuevo contexto, extensión o prioridad luego de finalizada la ejecución de script.
- 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 marcación.

La forma correcta de llamar a los scripts desde el plan de marcado es la siguiente porque el canal esta listo para manejar variables desde su creación.

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

```
exten => 123,2,AGI(mi_script.pl|argumentos)
```

Todos los scripts que se van a utilizar deben estar localizados en la ruta por defecto en `/var/lib/asterisk/agi-bin`.

2.2. VoiceXML

2.2.1. Objetivos del VoiceXML.

La tecnología VoiceXML está diseñada para crear diálogos de audio que presenta voz sintetizada, audio digital, el reconocimiento del lenguaje hablado y de entrada DTMF, registro de entrada de habla, telefonía, y las conversaciones de iniciativa mixta que son en las que tanto el usuario como el computador pueden afectar la conversación, según la referencia [8]. Su principal meta es acercar las ventajas de la Web basada en el desarrollo y entrega de contenido a las aplicaciones de respuesta de voz interactiva, como se puede observar en la figura 2.3.



Figura 2.3: Representación del intercambio de datos entre Asterisk y AGI.

2.2.2.Aspectos de una aplicación VoiceXML.

Una aplicación VXML esta constituida por un conjunto de etiquetas dialog states, según la referencia [9]. El usuario siempre se encuentra dentro de un dialog ya que a través de ellos se pueden saltar a otro dialog haciendo uso de un URL.

Hay dos tipos de dialog: forms y menus.

Una etiqueta form presenta y recoge información de voz o DTMF.

Una etiqueta menú presenta al usuario diferentes opciones y permite la transición a otros dialog.

2.2.3.Gramáticas de VXML.

Para usar el reconocimiento de voz en VXML podemos utilizar los siguientes ASR: Lumenvox, Verbio, Vtech, VoiceInteraction y uniMRCP ya que son los únicos soportados por esta tecnología, según la referencia [10].

Además de esto VXML nos ofrece etiquetas para crear diálogo de voz. Cuando un usuario empieza a utilizar vxml se crean etiquetas de raíz que toman el nombre de etiquetas grammar.

En una etiqueta grammar se establecen las palabras y declaraciones que serán validas como respuestas. Cabe indicar que estas declaraciones podrían ser desde una lista de posibles palabras hasta una frase compleja.

Una vez que se ha realizado el reconocimiento, la etiqueta grammar retorna variables a la aplicación VXML en curso.

A continuación mostramos un breve ejemplo de VXML, figura 2.3, en donde podemos comprender un dialogo entre el ordenador y una persona, según la referencia [11]. La interacción entre ellos es la siguiente:

Ordenador: Elija uno de los siguientes: Deportes; Tiempo; Noticias.

Persona: Astrología.

Ordenador: No comprendo este mensaje (esto se debe a que el usuario no escogió una de las opciones válidas).

Ordenador: Elija uno de los siguientes: Deportes; Tiempo; Noticias.

Persona: Deportes.

Ordenador: (Se dirige a <http://www.ejemplo.com/deportes.vxml> y continúa).

```
<?xml version="1.0"?>
<vxml version="2.0">
<menu>
  <prompt>
    Elija uno de los siguientes: <enumerate/>
  </prompt>
  <choice next="http://www.ejemplo.com/deportes.vxml">
    Deportes
  </choice>
  <choice next="http://www.ejemplo.com/eltiempo.vxml">
    El Tiempo
  </choice>
  <choice next="http://www.ejemplo.com/noticias.vxml">
    Noticias
  </choice>
  <noinput>Por favor diga uno de ellos<enumerate/></noinput>
</menu>
</vxml>
```

Figura 2.4: Ejemplo de reconocimiento de voz usando VXML [11]

2.3. Base de Datos Mysql.

Mysql es un Sistema Gestor de Bases de Datos (SGBD) que fue creado por la empresa sueca Mysql AB.

Este gestor de bases de datos es el más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de una gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

2.4. Software de Reconocimiento de Voz

2.4.1. Sphinx

Sphinx es el término que describe a un conjunto de sistemas de reconocimiento de voz continua. Además incluye una serie de reconocedores de voz entre los cuales podemos encontrar Sphinx2, Sphinx3, Sphinx4 y un entrenador de modelo acústico cuyo nombre es SphinxTrain [14].

2.4.2. Modelo Acústico.

Las versiones más recientes de Sphinx nos proporcionan varios modelos acústicos de calidad. Actualmente existen modelos acústicos en inglés-EE.UU., español, francés y chino los cuales han sido optimizados cuidadosamente para mejorar los resultados en el reconocimiento de voz. Existen modelos acústicos para micrófonos y para el habla en un teléfono. [15]

Un modelo acústico reconoce los sonidos naturales a un lenguaje específico. Para elaborar un modelo acústico eficiente en un idioma se necesita analizar más de 2000 horas de grabaciones de nativos leyendo

diversos textos, aunque basta con 140 horas de grabaciones para elaborar uno de mediana fidelidad.

La elaboración de un modelo acústico es un proceso complejo que toma como mínimo seis meses teniendo un conocimiento avanzado de fonética, lingüística, y conceptos afines. Por esta razón son pocos los modelos acústicos libres de buena calidad, y se da el caso de motores de reconocimiento de voz libres que usan por defecto un modelo acústico eficiente pero privativo.

2.4.3. Modelo de Lenguaje

Las fuentes de conocimiento sintáctico, semántico dan lugar a un modelo de lenguaje. Este modelo se encarga de estudiar la lógica del lenguaje, 'entendiéndolo' para calcular probabilidades de qué palabra sucede a cuál al decir una oración. El objetivo de este modelo es ahorrar tiempo al detectar palabras teniendo una lista de palabras que probablemente deberían seguirla, por ejemplo, al decir un artículo, se espera que la siguiente palabra sea un sustantivo [16].

2.5. Comprensión del Lenguaje Natural – NLU

La comprensión del lenguaje natural mejor conocido como NLU, es la ciencia de hacer que un ordenador entienda el lenguaje natural de las personas. El ordenador de reconocimiento podrá estar configurado de tal manera que sea capaz de procesar frases en cualquier idioma [17].

El hombre ha intentado avanzar en la tecnología de los NLU, pero no ha tenido éxito, ya que el ordenador de reconocimiento necesitaría tener un amplio conocimiento del mundo real. Esto se debe a que en los idiomas existen palabras que se pronuncian igual pero que poseen un significado diferente.

En el idioma español existen las palabras homófonas, que son un ejemplo claro de palabras ambiguas debido a que se escriben y pronuncian igual pero su significado es distinto.

2.6. Festival

Festival es un sistema de síntesis de habla libre que convierte el texto normal en lenguaje hablado. Fue desarrollado por el Centro de para la investigación de Tecnología hablada de la universidad de Edimburgo [31].

Actualmente una de sus grandes ventajas es que posee una buena calidad

de voz y pronunciación. Festival encuentra disponible en tres idiomas: ingles, español y galés [18].

Festival posee un diccionario de palabras, y cuando alguna palabra no se encuentra en allí, el tiende a deletrear. Cabe indicar que este software no soporta pdf, sino únicamente archivos de texto.

2.7. Perl

Perl (Practical Extraction and Report Language) es un lenguaje de programación para la administración de aplicaciones CGI para la web. Es un lenguaje libre de uso. Antes estaba muy asociado a la plataforma de Unix, pero en la actualidad se puede usar en Windows [19].

Los lenguajes de programación como c, sh, awk y sed, pero esta enfocado a ser más fácil y practico que los 3 ultimos. Es por ello que un programador que haya trabajado en c, se familiaria rápidamente con Perl.

2.8. PHP

PHP se define como un lenguaje de programación para la creación rápida de contenidos dinamicos de un sitio web. Cabe indicar que php en un lenguaje que se utiliza para programar aplicaciones del lado del servidor, mas no del lado del cliente.Hoy en dia php posee un lenguaje

multiplataforma y es de código abierto [20].

2.9. Smarty

Smarty es un motor de plantillas para php. Básicamente es una herramienta para poder separar la lógica y el contenido de la presentación. Con esto evitaríamos tener embebido código html en nuestros scripts de php. [21].

2.10. Softphone

Un softphone es un software que simula un teléfono convencional. En la actualidad existen muchos softphones entre los cuales tenemos a Zoiper, X-lite que son los más reconocidos ya que se los pueden instalar en Windows, Linux, y Mac. Normalmente este software se emplea en un entorno de Voz sobre IP [22].

CAPÍTULO 3

3. ESPECIFICACIONES Y HERRAMIENTAS PARA LA SOLUCIÓN.

3.1. Esquema General de la Solución

Se puede resumir el modelo completo de esta solución como se muestra en la figura 3.1.

Cuando el usuario llama a la extensión 123, que es el número para alcanzar a nuestro IVR, como se ve en el `extensions.conf` (Anexo 4) lo redirecciona al script AGI de bienvenida (Anexo 5.1). Desde este script se llama a otros scripts AGI dependiendo de variables de canal y registros en la base de datos. De esta manera se pasará de un módulo a otro en la llamada, por ejemplo, de Autenticar Cliente a Menú principal, y luego al submenú que seleccione el usuario. Cada módulo se explica a fondo analizando su esquema en la sección 3.7.

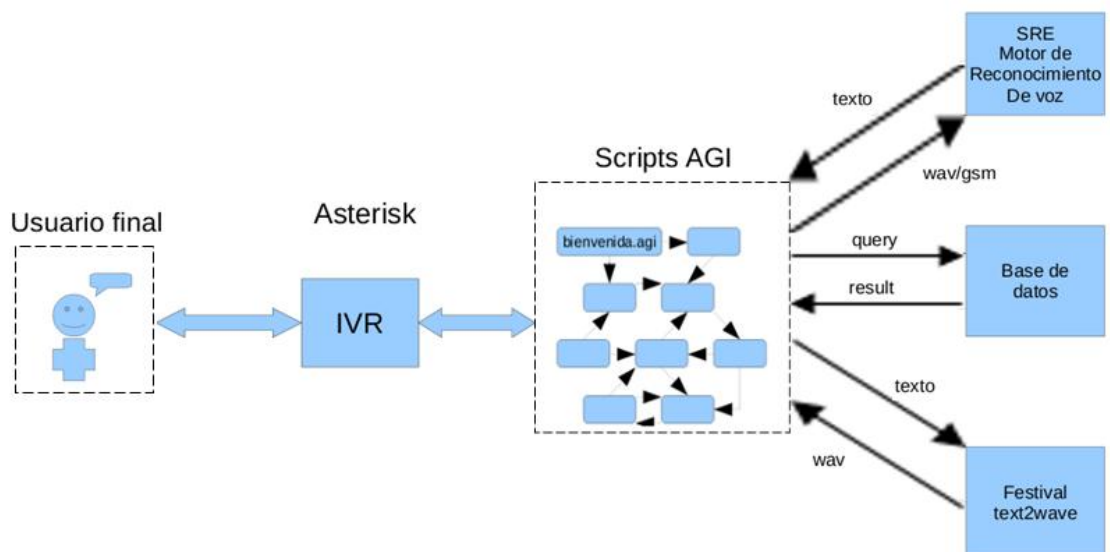


Figura 3.1: Esquema general de la solución

Un script AGI puede requerir una conexión a la base MySQL para realizar registros, actualizaciones o consultas. Esto lo hará ejecutando un query y recibiendo un arreglo con los resultados.

Para presentar información al usuario o para solicitarle el ingreso de datos, se utilizará un conversor de texto a voz (TTS). Con esta tecnología se facilita la actualización y personalización del texto que se leerá al usuario evitándonos el proceso de grabar nuevos archivos de audio con la voz de la misma persona. El conversor a utilizar será la herramienta text2wave de Festival, el sistema de síntesis de habla que viene por defecto en casi todas

las versiones actuales de Linux. Text2wave recibe texto y genera un archivo temporal wav que se 'dirá' utilizando la función stream_file() del AGI de Asterisk.

Para escuchar el habla del usuario y reconocer lo que ha dicho se utilizará el motor de reconocimiento de voz (SRE) Sphinx2. Desde el script AGI de Asterisk se graba lo que dice el usuario en un archivo wav temporal que es convertido en texto por la función asr(). En el mismo script se utiliza esta respuesta para decidir qué acción realizar a continuación y de esta manera se navega en los menús.

3.2. Especificaciones del servidor

El procesador de nuestro servidor de Asterisk, Sphinx y Festival es un Intel(R) Core(TM)2 Duo T5550 de 1.83GHz con 3GB de RAM y un disco duro de 250 GB. Se detallan más especificaciones de procesamiento en el anexo 1.1 y de memoria en el anexo 1.2. Con una rapidez de procesamiento mayor a 1GHz o cerca de esta cantidad, se podría soportar entre 20 y 60 llamadas concurrentes al servidor Asterisk. [23]

El servidor de este sistema tiene instalado el sistema operativo Linux, con la distribución Centos en la versión 5.2. Revisamos la versión del sistema

operativo con este comando:

```
[root@localhost ~]# uname -a && cat /etc/*release
Linux localhost.localdomain 2.6.18-238.19.1.el5 #1 SMP Fri Jul 15 07:32:29
EDT 2011 i686 i686 i386 GNU/Linux
Red Hat Enterprise Linux Server release 5.2 (Tikanga)
```

3.3. Modelo de la base de datos

Utilizamos MySQL como sistema de gestión de base de datos porque es relacional, de código abierto y tiene popularidad. La versión en la que se ha desarrollado la base de este sistema es la 5.0.77.

```
mysql> SHOW VARIABLES LIKE "%version%";
```

Variable_name	Value
protocol_version	10
version	5.0.77
version_bdb	Sleepycat Software: Berkeley DB 4.1.24: (January 29, 2009)
version_comment	Source distribution
version_compile_machine	i686
version_compile_os	redhat-linux-gnu

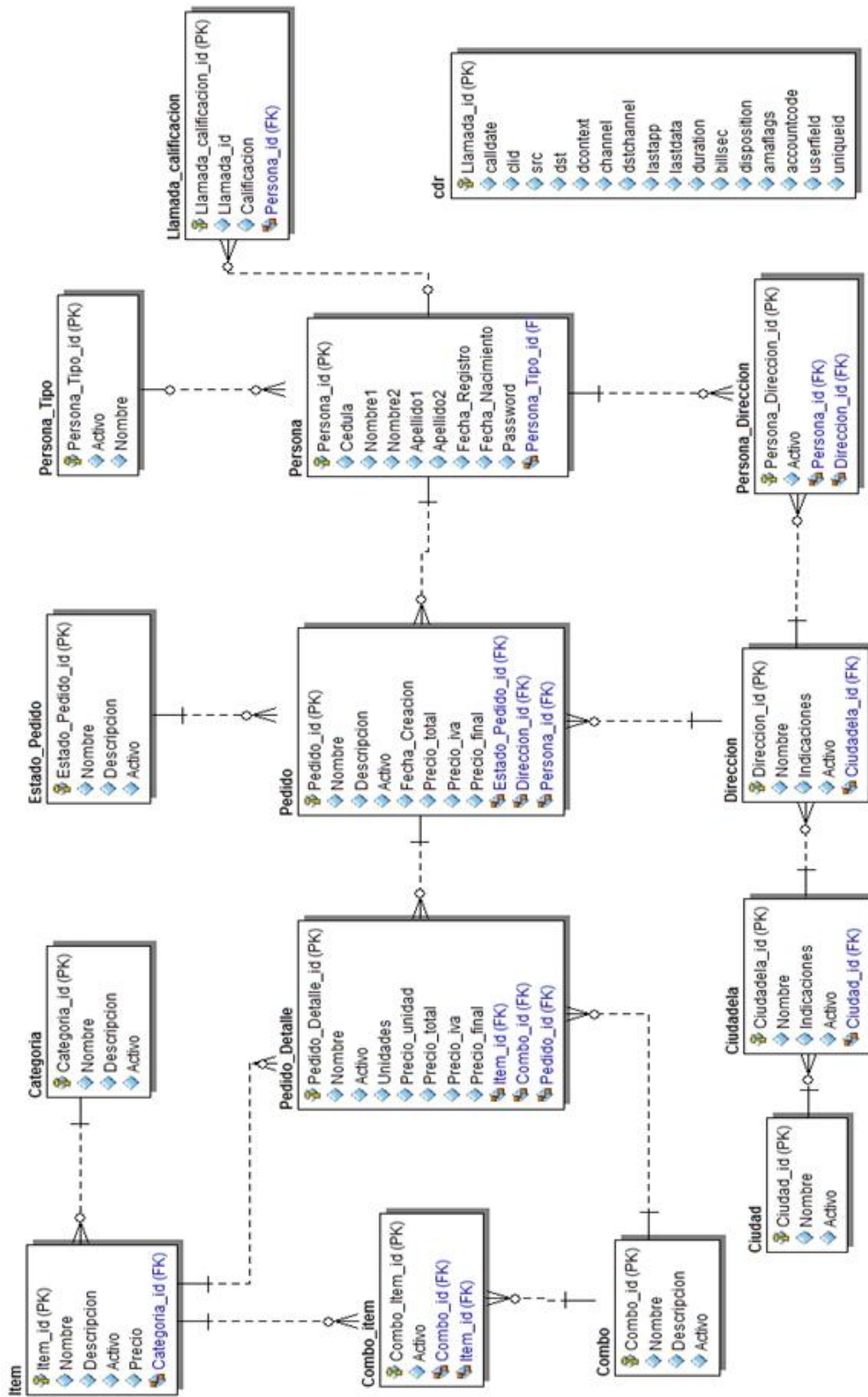


Figura 3.2: Modelo físico de la base de datos.

Breve descripción de las tablas

Persona_Tipo

Esta es una tabla de tipo catálogo para configurar los diferentes roles de los usuarios del sistema, que al momento son Cliente y Empleado. Todo usuario que se registre por medio del IVR será guardado como cliente porque a ellos está dirigido el automatizador de llamadas. El tipo de persona Empleado sería necesario para acceder a la aplicación interna de los empleados del restaurante y poder manipular el estado de los pedidos.

Actualmente se crea a los empleados directamente en la base pero podría agregarse un módulo de creación de personas, posiblemente para otro tipo de usuario, por ejemplo Gerente. Así se crearían roles para distintos permisos llegando hasta SuperAdministrador.

Persona

Para registrar a una persona en nuestro sistema basta ingresar el número de cédula y configurar una contraseña, guardando el campo Fecha_registro en el proceso. En este proyecto asumimos que nuestra aplicación tiene acceso a una base universal con los datos necesarios para la facturación. Los demás campos en esta tabla no se llenan activamente en el registro, sino que se asume que son extraídos de esta otra base para luego en el

reporte de pedidos contar con el nombre completo y dirección del usuario.

Direccion, Persona_Direccion, Ciudadela y Ciudad

Estas tablas están creadas de forma que se soporte de forma normalizada el caso de que una persona tenga varias direcciones o requiera la entrega de su pedido en una nueva dirección. Ciudadela y Ciudad existen para agilizar el proceso de toma de una dirección manualmente.

Cdr

Este es por defecto el registro de detalle de llamadas de Asterisk. Los campos que contiene son:

src: Identificación del caller

dst: Extensión de destino

dcontext: Contexto de destino

channel: Canal utilizado

dstchannel: Canal de destino

lastapp: Última aplicación correctamente ejecutada

lastdata: Últimos datos de aplicación (argumentos)

start: Fecha en la que se comienza la llamada

answer: Fecha en la que se responde la llamada.

end: Fecha de fin de la llamada

duration: Tiempo total de la llamada desde que se marca hasta que se

cierra.

billsec: Tiempo total de la llamada desde que se contesta hasta que se cierra.

disposition: En qué estado quedó la llamada. Puede ser ANSWERED, NO ANSWER, BUSY, FAILED

Llamada_calificacion

Si el cliente opta por colgar diciendo 'Chao' en cualquier estado de la llamada (a excepción de Seleccionar Items), la llamada se cerrará pero antes se pedirá una calificación del 1 al 10 del nivel de satisfacción de la llamada. No es obligatorio que el cliente responda. En esta tabla se guardará esta calificación junto al uniqueid de la llamada y el ID de la persona en caso de estar autenticada.

Categoría

Tabla de tipo catálogo que guarda todas las categorías de los productos que venda el restaurante, con la opción de marcar como no activa a las que en algún momento no deban ser ofrecidas en el menú de Categorías del IVR.

El campo Descripción es el campo que va a leer el conversor de texto a voz (TTS), así que si el nombre de la categoría es difícil de pronunciar para el TTS, este debe ser simplificado, caso contrario debe ser igual al nombre. Si

en un registro no está lleno el campo Descripción, el IVR lo ignorará y el TTS no leerá esta categoría.

Ítem

Guarda el catálogo de todos los productos que ofrece el restaurante, indicando su categoría. Los campos Activo y Descripción funcionan de igual manera que en Categoría. El precio guardado en esta tabla es previo al IVA.

Combo y Combo_item

En un combo se agruparía distintos ítems con precios diferentes a sus precios individuales. Este menú no se desarrolló por no aumentar la complejidad del IVR actual.

Pedido_Detalle

Un detalle de pedido es la selección de un ítem o combo y la cantidad deseada. Los campos Nombre y de precios se guardan aquí de manera redundante para mostrarse en un historial de pedidos. Se debe soportar el caso de que un ítem o combo cambie de precio, entonces en esta tabla se mostrarán los precios existentes al momento de la toma del pedido, no los actuales que pueden ser mayores o menores. Todo detalle debe permanecer a un pedido.

Pedido

En esta tabla también se aplica la redundancia para soportar la variabilidad de los precios. Los precios guardados serán las sumas de los pedido_detalle activos correspondientes al pedido. El nombre y descripción ya no se están utilizando para narrar pedidos. Todo pedido debe corresponder a una Persona y debe tener una fecha de ingreso y un estado que lo describa.

Estado_Pedido

Los estados del pedido son: Creado, Confirmado, Comenzado, Despachado, Entregado y Pagado.

Cuando el cliente guarda un pedido al confirmarlo, este se ingresa a la base de datos como Confirmado. El estado Creado lo utilizamos en el desarrollo del proyecto para hacer pruebas ingresando datos directamente a la base. En la aplicación interna los empleados del restaurante tienen el módulo Actualización de estado de pedidos pendientes para cambiar de un estado de pedido al siguiente.

3.4. Instalación de Asterisk

Antes de empezar con la instalación de Asterisk, debemos preparar nuestro servidor. Para ello, verificaremos si nuestro servidor tiene instalado los

paquetes: bison, bison-devel, ncurses, ncurses-devel, libtermcap, libtermcap-devel, zlib, zlib-devel, openssl, openssl-devel, gnutls-devel, gcc, gcc-c++, newt, newt-devel, y libtool.

Podemos utilizar el comando `#rpm -q "paquete"` y `#yum install -y "paquete"` para verificar e instalar respectivamente. Además de esto debemos de sincronizar nuestro reloj del servidor con un servidor externo de sincronización de tiempo.

```
#yum install -y ntp && ntpdate br.pool.ntp.org && chkconfig ntpd on &&
service ntpd start
```

Una vez que hayamos realizado los pasos anteriores instalamos Asterisk. Comenzamos descargando el código fuente desde el portal web de Asterisk [31].

Desempaquetamos el código fuente de Asterisk y usamos los siguientes comandos:

```
# cd /usr/src/asterisk-version
# make clean
# ./configure
# make
# make menuselect
# make install
# make samples
# make config
```

Con estos pasos se da por terminada la instalación. Para la implementación

de este proyecto utilizamos la versión 1.8 de Asterisk. Para conocer detalles de la versión utilizamos el comando:

```
localhost*CLI> core show version  
Asterisk 1.8.3.2 built by root @ localhost.localdomain on a i686 running Linux  
on 2011-06-19 04:17:47 UTC
```

Esta versión está disponible desde el 17 de marzo del 2011 según [20].

3.5. Componentes necesarios para la aplicación web

3.5.1 Instalación de PHP

Para poder utilizar la aplicación web, debemos instalar PHP y Smarty. Con el siguiente comando verifico si está instalado php:

```
#rpm -q php
```

Si no tengo instalado php lo instado de la siguiente manera.

```
#yum install -y php
```

Si el proceso de descarga termina sin errores, la instalación ha sido exitosa.

La versión de php que tenemos instalada en nuestro equipo es la 5.1.6. Podemos conocer la versión de PHP con el comando:

```
[root@localhost ~]# php -v
PHP 5.1.6 (cli) (built: Nov 29 2010 16:47:37)
Copyright (c) 1997-2006 The PHP Group
Zend Engine v2.1.0, Copyright (c) 1998-2006 Zend
Technologies.
```

3.6. Vxml

La especificación de este proyecto sugería originalmente el desarrollo de la aplicación utilizando Vxml. Como se muestra en el marco teórico en un ejemplo de seleccionar secciones de un periódico 'hablando' con la aplicación Vxml, el código es simple y nos resulta fácil porque se parece al standard html que nos es bastante familiar.

No utilizamos Vxml para el desarrollo de los módulos de este sistema porque tuvimos problemas de instalación del VXI* VoiceXML Browser (de i6net) que no pudimos resolver. Probando con los ejemplos de las guías de la página de i6net los resultados fueron los siguientes:

Sí se ejecutaban los programas en Vxml al llamarlos desde las extensiones de Asterisk y seguían la rutina especificada como pedir una entrada de datos al usuario sin arrojar errores.

La conversión de texto a voz de los scripts Vxml era pésima, porque el volumen del ruido era mucho mayor al de la voz que apenas se diferenciaba. El reconocimiento de voz no nos funcionó. No es que fuera de mala calidad como en el caso del TTS, sino que parecía que no se lo estaba

llamando.

Estas irregularidades persistieron a pesar de hacer cambios en distintas configuraciones. No se logró identificar el problema porque no había errores en los logs ni al ejecutar los scripts para de esa forma guiarnos.

Para revisar que no sea un problema de falta de librerías o de configuraciones de Asterisk, luego probamos el Sphinx2 que finalmente sí pudo reconocer el habla en un canal Asterisk.

Nos decidimos por el Sphinx2 porque a pesar de no tener una integración nativa con Asterix, ésta sí es posible y encontramos varios casos de usuarios que han logrado esta integración utilizando un modelo cliente servidor y la extensión de Perl para Sphinx 2 Speech::Recognizer::SPX.

3.7. Sphinx2

El motor de reconocimiento de voz que se usa en este proyecto es Sphinx2, la versión original de Sphinx escrita en lenguaje C [24].

La versión 3 de Sphinx escrita en C++ y la versión 4, escrita en Java tienen una mejor fidelidad en el reconocimiento de voz, pero Sphinx2 es más rápido [25] [26]. Sphinx2 debe funcionar correctamente en Linux, Solaris, varias versiones de Unix, y en Windows usando Visual Studio y cygwin. Como Sphinx4 está escrito en Java, puede correr en múltiples plataformas.

Sphinx4 ha sido probado en: Solaris 9, Mac OSX 10.3.5, RedHat 9.0, Fedora Core 1, Microsoft Windows XP, y Windows 2000. [27]

La diferencia de Sphinx con respecto a otros SREs es que no requiere aprendizaje o entrenamiento para reconocer palabras sino sólo el diccionario de los términos a reconocer. El uso de RAM varía de acuerdo al tamaño del diccionario utilizado.

Aunque la versión 2 está considerada como obsoleta y de esta hay menos documentación existente en comparación con las siguientes versiones, es la que ha tenido más casos de éxito integrándose con Asterisk [28] y es la versión de la que hemos encontrado más discusión y ejemplos [29] por parte de usuarios en Internet. Por esta razón utilizaremos Sphinx2 para este proyecto y no sus versiones más actuales. Tener reconocimiento de voz y no poder utilizarlo en la llamada telefónica impediría completamente lograr el objetivo de este proyecto.

Para instalar Sphinx2 en nuestro servidor, seguimos los pasos de la guía [29] más algunas modificaciones.

1. Descargamos la versión 0.6 de Sphinx2 del link http://sourceforge.net/project/showfiles.php?group_id=1904.

2. Descomprimos el paquete y en la carpeta generada ejecutamos:

```
# ./configure --prefix=/usr/local/sphinx  
# make  
# make install
```

Esto instala Sphinx2 en /usr/local/sphinx

3. Descargamos los módulos de Perl para Sphinx Speech-Recognizer-SPX desde:

<http://search.cpan.org/CPAN/authors/id/D/DJ/DJHD/Speech-Recognizer-SPX-0.0801.tar.gz>

4. Descomprimos el paquete y en la carpeta generada ejecutamos:

```
# perl Makefile.PL --sphinx-prefix=/usr/local/sphinx  
# make  
# make install UNINSTALL=1
```

Esto instala el módulo de Perl en el mismo directorio.

5. Creamos un archivo de texto con las palabras a reconocer en cada línea y lo subimos al webservice Lmtool. [30].

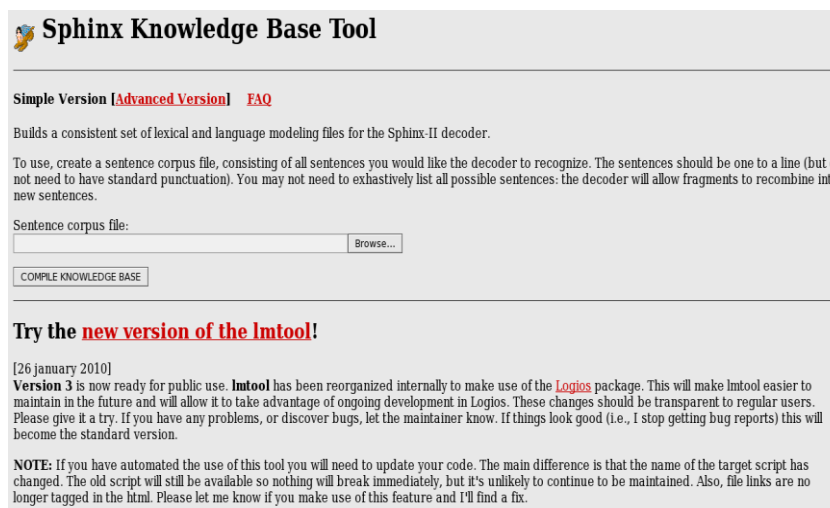


Figura 3.3: Webservice Lmtool.

Éste nos genera un diccionario, un paquete con los archivos .corpus, .dic, .lm, .sent .sent.arpa, .token y .vocab (Anexo 2.3).

6. La carpeta generada al descomprimir este paquete debe ser colocada en la ruta de los modelos de lenguaje /usr/local/sphinx/share/sphinx2/model/lm/
7. Descargamos un modelo acústico para Sphinx2 desde <http://www.speech.cs.cmu.edu/sphinx/models/hmm/communicator-2000-11-17-2.tgz>
8. En la carpeta generada al descomprimir este paquete ejecutamos el comando:

```
#mvsphinx_2_format
/usr/local/sphinx/share/sphinx2/model/hmm/communicator
```

9. Descargamos el `sr_server.dat` (Anexo 2.1) desde http://pbxinaflash.net/source/sphinx/sr_server y lo colocamos en la ruta `/usr/local/bin`. 'confirm' es el nombre del modelo de lenguaje/diccionario que generamos en el paso 5.

10. Para comprobar que Sphinx2 se haya instalado correctamente, podemos probar con archivos wav. Para esto colocamos en `/usr/local/bin` el `sr_client.dat` (Anexo 2.2) descargado de:

```
# wget http://www.syednetworks.com/asterisk-sphinx.rar
```

11. Comprobar que tanto el `sr_server.dat` y el `sr_client.dat` sean ejecutables.

12. Para levantar el Sphinx ejecutamos estos comandos en consola:

```
#export LD_LIBRARY_PATH=/usr/local/sphinx/lib
#cd /usr/local/bin
#sr_server.dat &
```

13. Para probar el reconocimiento de voz con archivos pregrabados ejecutamos:

```
#!/usr/local/bin/saludo.wav
```

Esto nos debería retornar 'Result: SALUDO'

3.8. Esquemas de los scripts AGI

Utilizamos el AGI de Asterisk en Perl para ejecutar en el canal los distintos módulos que conforman la llamada. Además logramos comunicarnos con la base, setear variables de canal y atender la entrada de datos por voz o DTMF. La versión de Perl utilizada en este proyecto es la 5.8.8. Para conocer detalles de la versión utilizamos el comando:

```
[root@localhost ~]# perl -v  
This is perl, v5.8.8 built for i386-linux-thread-multi. Copyright 1987-2006,  
Larry Wall
```

A continuación se resumirá de manera breve el comportamiento de cada módulo. En todos los que esperan ingreso de un comando de voz por parte del usuario, diciendo 'Chao' se llega al módulo 'Chao: Cerrar llamada'.

3.8.1. Opciones del menú principal

Las opciones del menú principal cambian dependiendo si el cliente está autenticado o no y en el primer caso también afecta el que tenga pedidos pendientes (no terminados) o detalles pendientes por confirmar:

Menú (no autenticado)	Menú (autenticado)
Menú de categorías	Menú de categorías

Autenticarse

Confirmar pedido (Si hay detalles de pedido por confirmar)

Más Información

Revisar estado del pedido (si existen pedidos no pagados)

Más Información

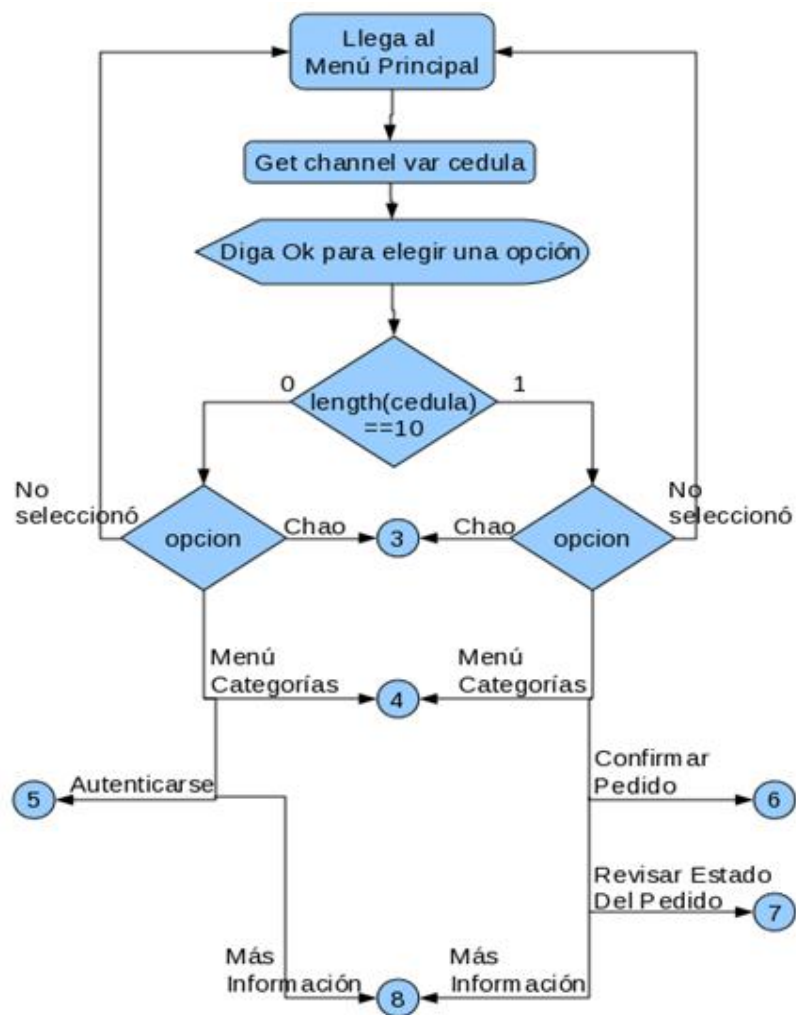


Figura 3.4: Menú principal.

3.8.2. Bienvenida

Este script contesta la llamada y redirecciona al módulo 5, 'Autenticar/Registrar Cliente'. La subrutina 'Get channel var cedula' iba a tomar la variable de canal que contiene al identificador del caller y con ella consultar a la base el número de cédula de la persona que llama. Esto sería correcto si no se diera el caso de que distintos usuarios realicen llamadas desde un mismo teléfono. Este caso no se consideró al modelar la base de datos. En esta subrutina siempre tomará una variable de canal no seteada, por lo que irá al módulo Autenticar Cliente.

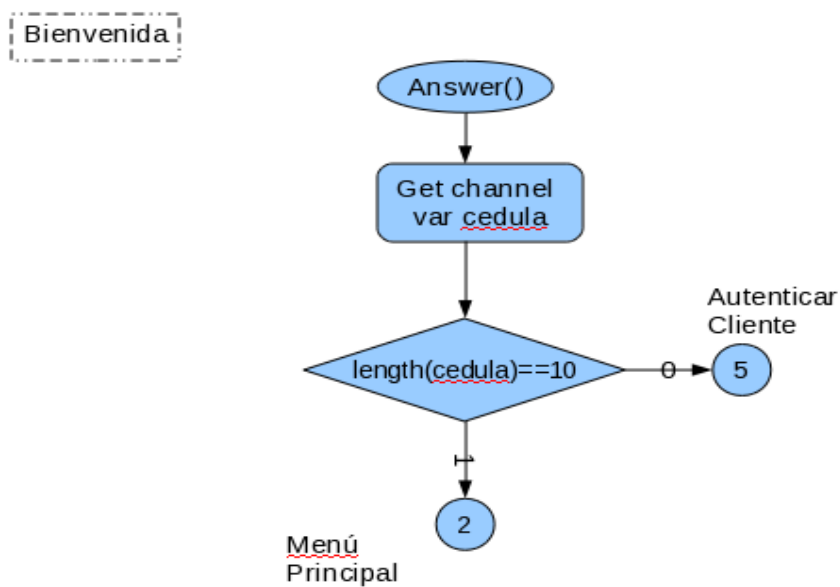


Figura 3.5: Bienvenida

3.8.3. Autenticar Cliente

Al llegar a este script se solicita el ingreso del número de cédula de la persona a autenticar. Cuando lo ingresa, si existe en la base de datos le revisa que ingrese su contraseña. Si el usuario no está autenticado, se le pregunta si se quiere registrar en el sistema. En el caso positivo, se le pide que ingrese una contraseña de entre cuatro y ocho dígitos para registrarlo en la base.

Sólo en el caso de que el usuario se autentique correctamente o se registre por primera vez se guardará la cédula como variable de canal para recordar en los otros módulos que hay un usuario autenticado.

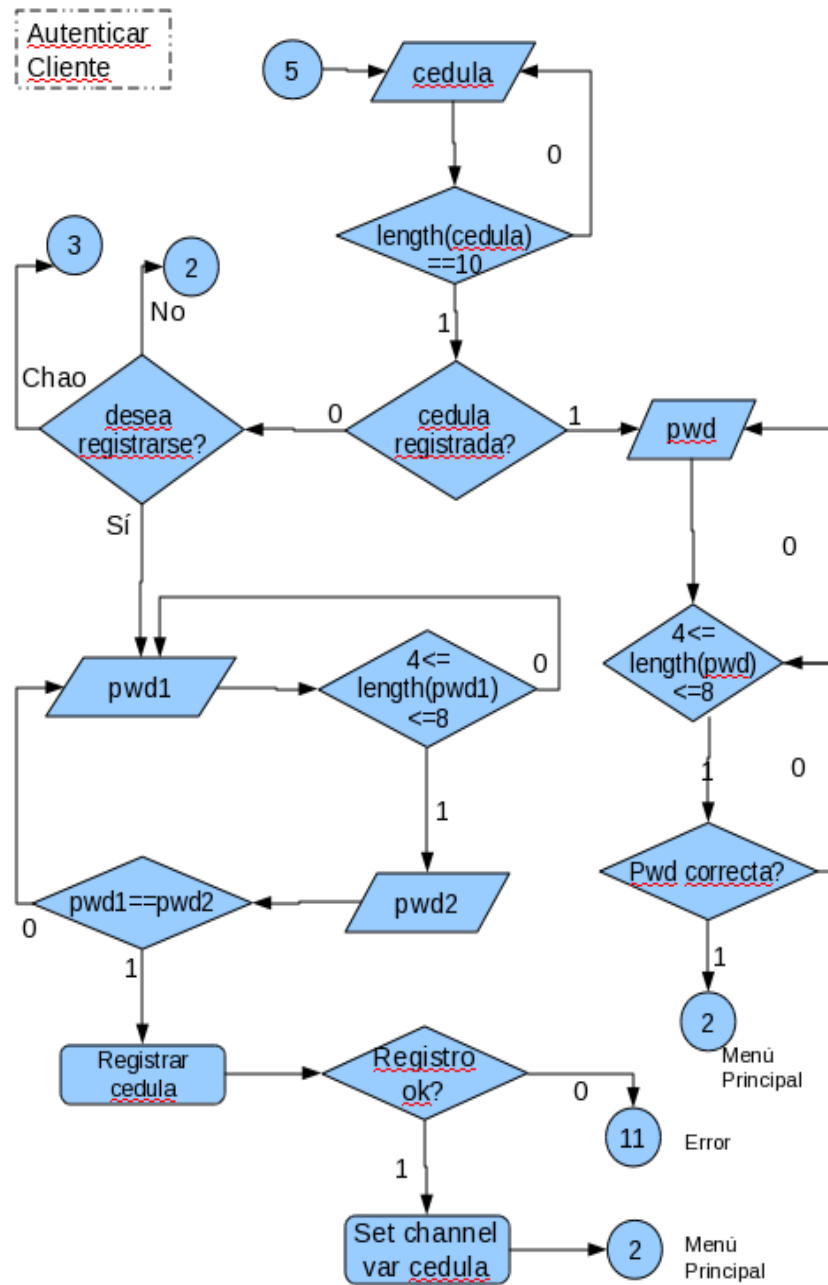


Figura 3.6 Autenticar Cliente

3.8.4. Menú Principal

Como se puede revisar en la figura 3.7, siempre se informará al usuario que ha llegado al menú principal y luego se armará el menú como fue explicado en la sección 3.7.2. Se recorre cada opción del menú leyéndola y se ofrece al usuario un máximo de intentos para decir Ok si quiere acceder al menú. Si recibe 'Cancelar' repite el menú principal.

Si se acaban las opciones del menú y no se ha seleccionado ninguna, se le pregunta al cliente si desea repetir el menú principal y también se ofrece un límite de intentos para contestar. Si el módulo recibe el comando 'Cancelar' se repite el menú principal."

3.8.5. Menú de Categorías

Al comenzar este módulo se informa al usuario que ha llegado al menú de Categorías y se consulta a la base el listado de categorías activas. Aunque no se debería dar este caso, si no hay ninguna disponible se informa "Al momento no hay categorías activas" y se redirecciona al usuario al menú principal. Esto se puede revisar en la figura 3.8.

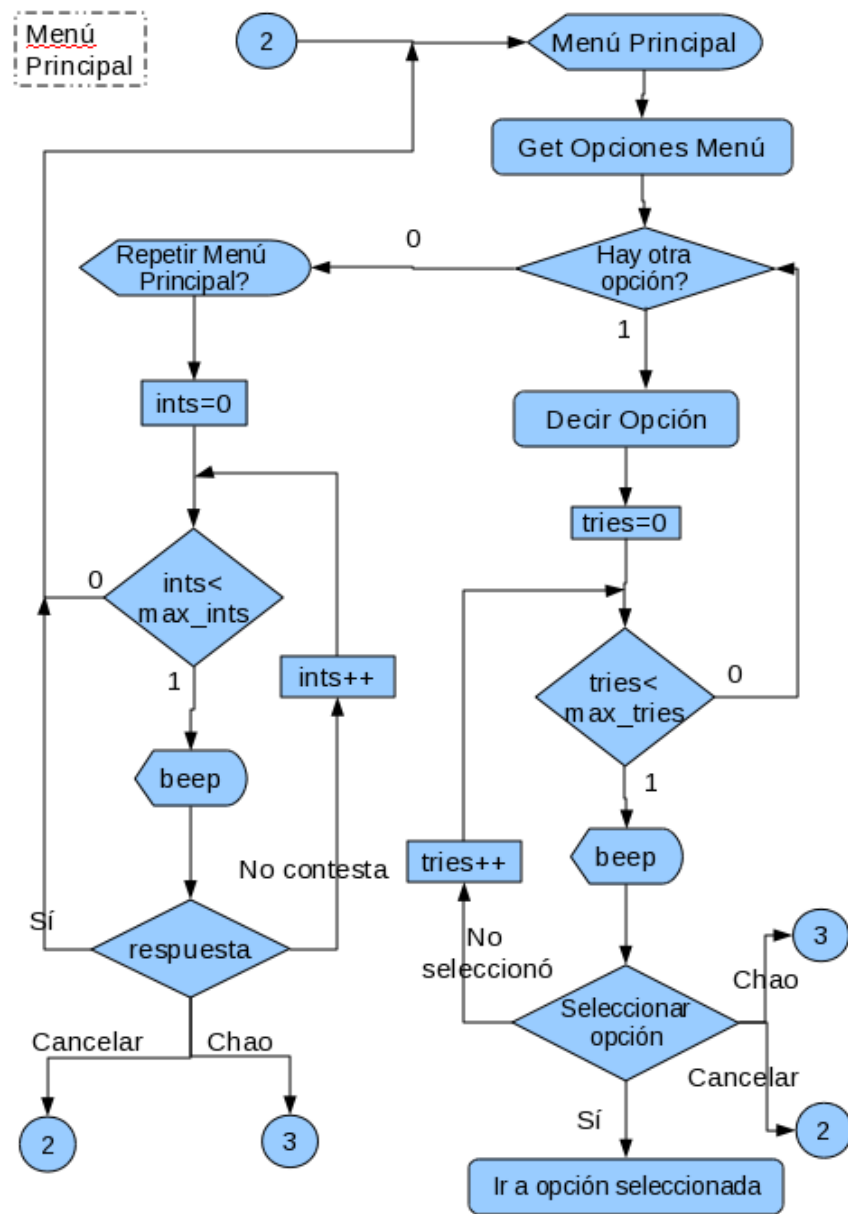


Figura 3.7: Menú principal

Normalmente habrá un listado de categorías que recorrer; en cada una se narra el nombre de la categoría como está especificado en el campo Descripción. Luego de nombrarla, se ofrece al cliente un límite de intentos para acceder a los ítems de la categoría diciendo Ok o sí, en cuyo caso setea la variable de canal `selected_categoria_id` con el ID de la categoría seleccionada y se ejecuta el módulo `select_item`.

Si no selecciona ninguna categoría, se revisa si existe la variable de canal `num_selected_items` que es el número de líneas que contiene el archivo temporal de detalles. Si esta variable es mayor a cero, quiere decir que tiene detalles para confirmar realizar un pedido así que se lo redirecciona al módulo `Confirmar Pedido`.

Si no selecciona ninguna categoría y no ha formado detalles se le da intentos al cliente para responder 'Sí' si quiere repetir el menú de categorías o 'Cancelar' para volver al menú principal. Si no selecciona nada, se repite el menú principal.

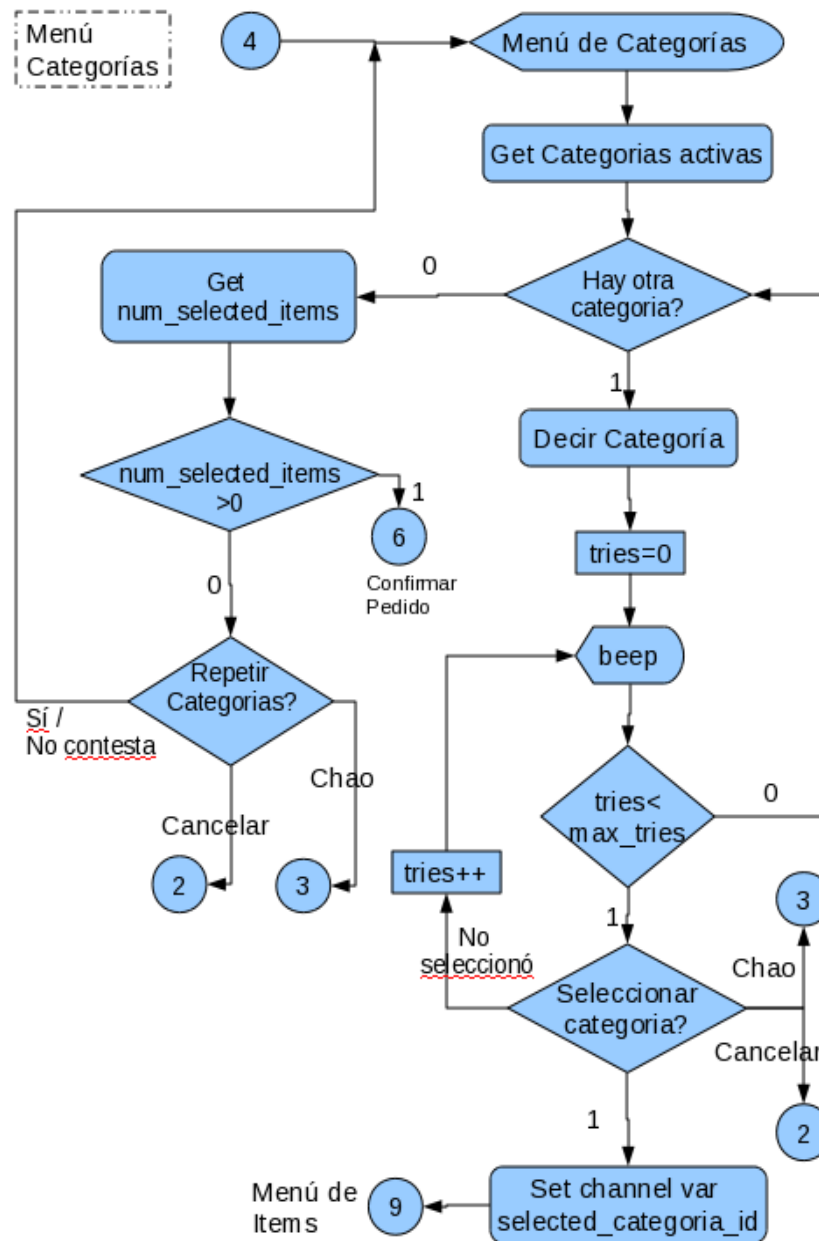


Figura 3.8: Menú categorías.

3.8.6. Menú de ítems

Como se puede ver en la figura 3.9, al comenzar este módulo revisamos la variable de canal `selected_categoria_id`, que debería ser mayor a cero. Si no lo es, redirecciona al usuario al módulo Error (no se debería dar este caso). Si el ID de categoría es correcto, se consulta a la base el total ítems activos correspondientes. Si el total de ítems de la categoría no es mayor a cero, se redirecciona al cliente al menú de categorías.

Si el total es mayor a cero significa que la categoría sí tiene ítems activos, así que se los consulta a la base. Ahora se verifica si el usuario está autenticado; si lo está se ejecuta el módulo Presentar ítems para seleccionar, sino se ejecuta el módulo Listar ítems.

3.8.7. Listar ítems

Al llegar a este módulo se le comunica al usuario que ha llegado al menú de ítems. Luego se nombra todos los ítems obtenidos en la consulta de ítems activos por `selected_categoria_id`. Al terminar de listar los ítems se redirecciona al usuario al menú de categorías. Revisar figura 3.10.

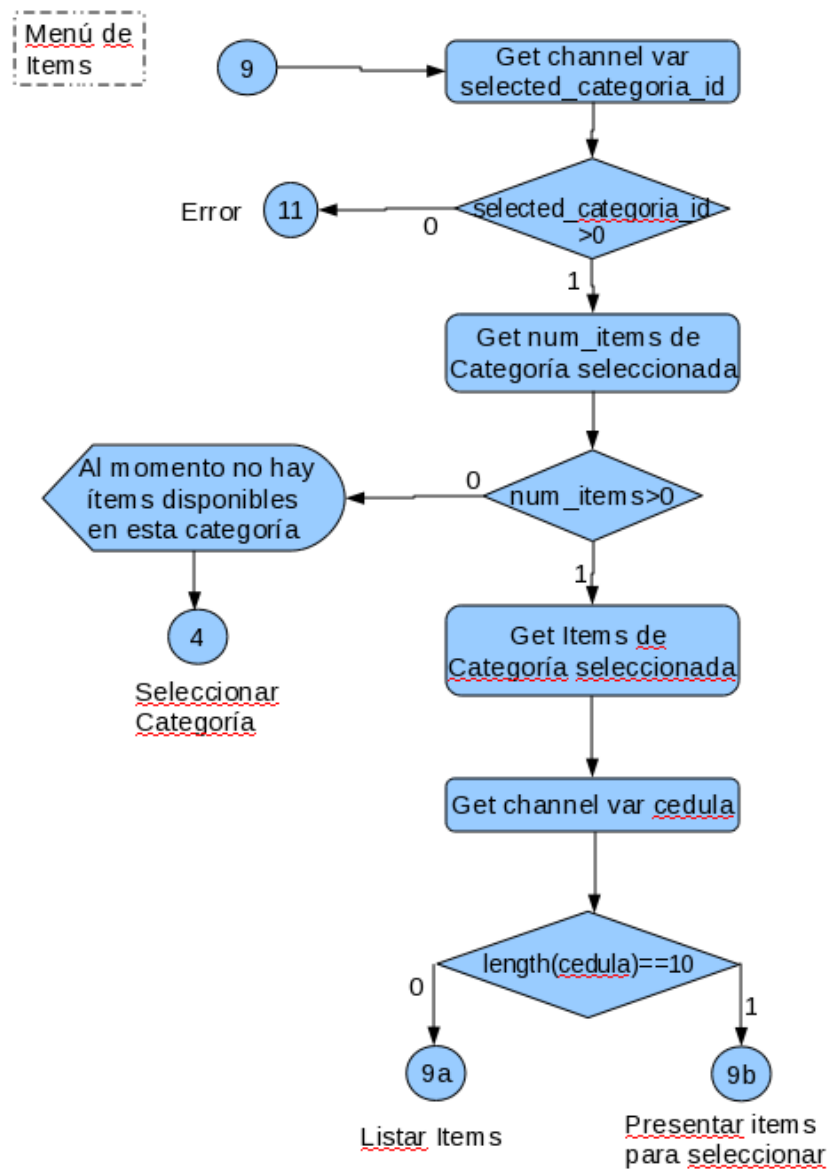


Figura 3.9: Menú de Items.

Listar
Items

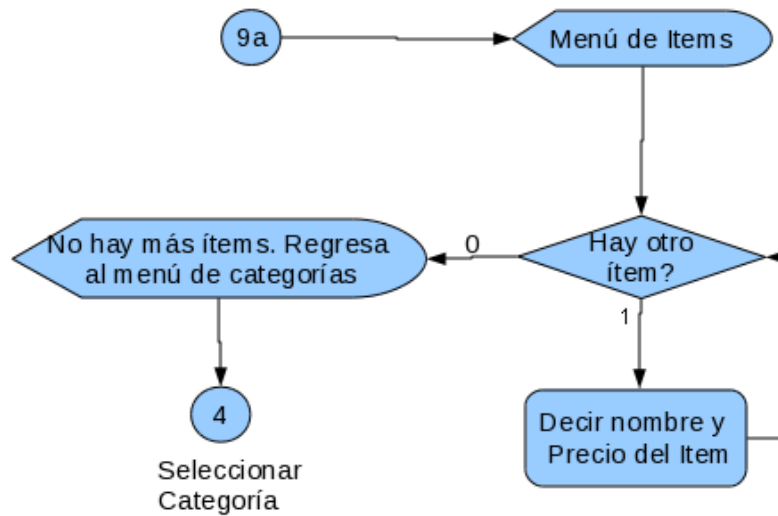


Figura 3.10: Listar Items.

3.8.8. Presentar ítems para mostrar

Al llegar a este módulo se informa al usuario que debe ingresar las siguientes opciones por el teclado alfanumérico (nuestro diccionario no incluye números):

- si quiere seleccionar un ítem debe ingresar la cantidad que desea comprar.

Esta cantidad debe ser menor o igual a 15.

- si quiere confirmar el pedido presionar 98.

- si quiere volver al menú de categorías, presionar 99.

Si se presiona 98, antes de ejecutar el módulo Confirmar Pedido se debe verificar que haya ítems seleccionados.

Al ingresar la cantidad de ítems que desea comprar se narra al usuario lo elegido, por ejemplo: 'Elegió 2 hamburguesas vegetarianas'. Luego se guarda la selección en una línea del archivo temporal tmp_file_noconf en el formato "item_id,cantidad". Si este archivo estaba vacío, se setea el nombre del archivo temporal como la variable de canal tmp_file_noconf.

Si se ingresa una cantidad menor a 1 y mayor a 15 se narrará este ítem otra vez (desde cero intentos). Cuando se ha presentado todos los ítems se redirecciona al usuario al menú de categorías.

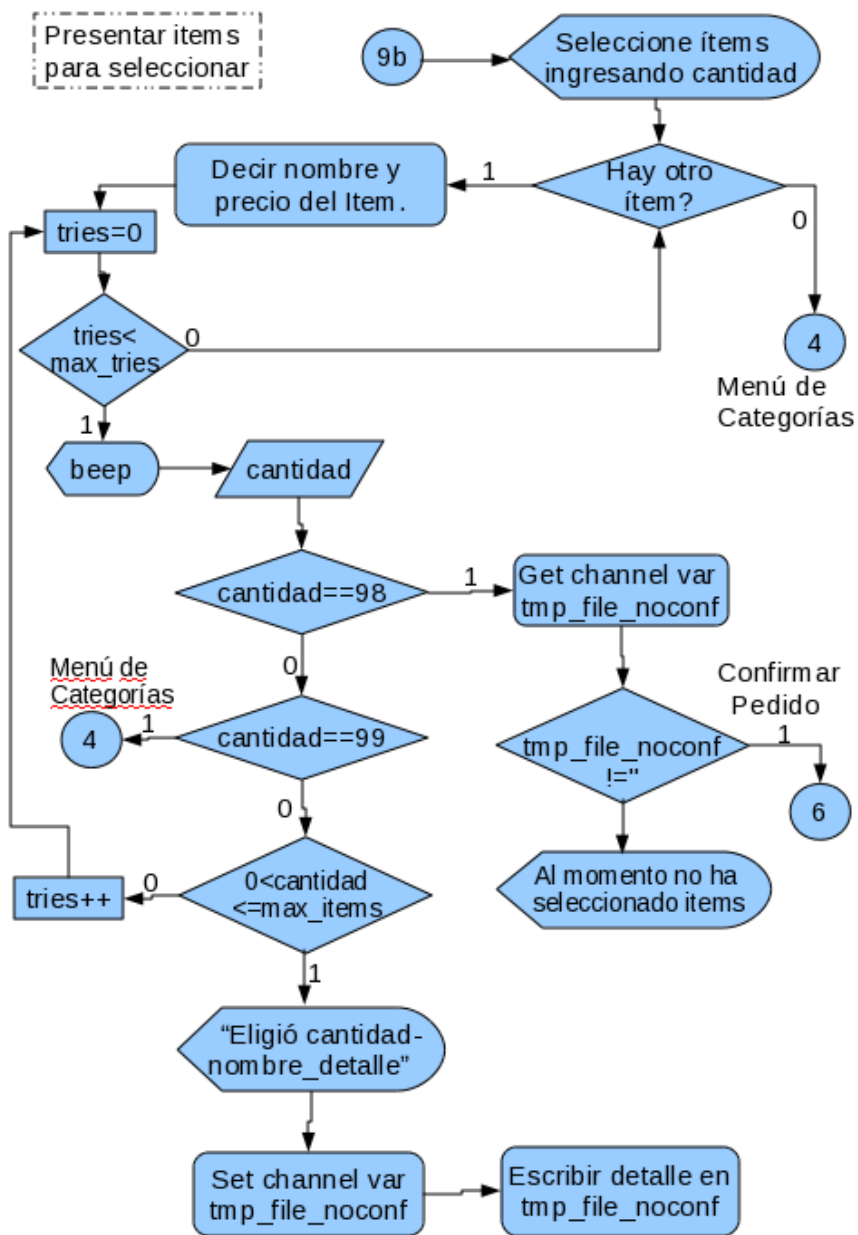


Figura 3.11: Presentar ítems para seleccionar.

3.8.9. Confirmar Pedido

Al llegar a este módulo se le comunica al usuario que debe decir No para ignorar un ítem y que éste no sea considerado en el pedido.

Se obtiene todos los detalles seleccionados parseando el archivo temporal e inicializamos en 0 una bandera pedido_revisado. Se consulta en la base los datos de los detalles seleccionados obtenidos del archivo temporal. Además se inicializa vacío un arreglo de detalles confirmados.

Se recorre el arreglo de detalles consultados a la base diciendo la cantidad, nombre y precio final de cada detalle. Por cada detalle se inicializa la bandera 'negado' en cero y se ofrece un máximo de intentos para negarlo. Si luego de terminarse los intentos el detalle no ha sido negado, éste se agrega al arreglo de detalles confirmados.

Si se ha terminado de narrar los detalles que habían sido seleccionados se cierra el archivo temporal y se comunican los precios base y final. Se le ofrece al cliente un número de intentos para contestar si quiere guardar el pedido. Si dice Cancelar, regresa al menú principal, si dice Ok se guarda el pedido en la base, y si dice No se vuelve a narrar todos los detalles del archivo temporal de detalles seleccionados.

Al guardar el pedido se obtiene su ID para informárselo al cliente, se elimina el archivo temporal de pedidos seleccionados y se blanquea la variable de canal `tmp_file_noconf`. El cliente es redireccionado al menú principal. Revisar la figura 3.13.

3.8.10. Revisar estado de pedidos pendientes

Al ejecutar este módulo se consulta a la base todos los pedidos pendientes activos. Se recorre este arreglo narrando el ID del pedido, su precio final y fecha de confirmación preguntando al cliente si quiere acceder a los detalles del pedido.

Si el cliente dice Ok, se setea la variable de canal `pedido_id` y se ejecuta el módulo Detalles del pedido.

Si el cliente no desea revisar los detalles de ningún pedido y ya se ha terminado el listado de pedidos, ofrecerá un máximo de intentos para contestar si se quiere repetir el listado. Si se acaba los intentos el cliente no contesta a esta pregunta, será redireccionado al menú principal.

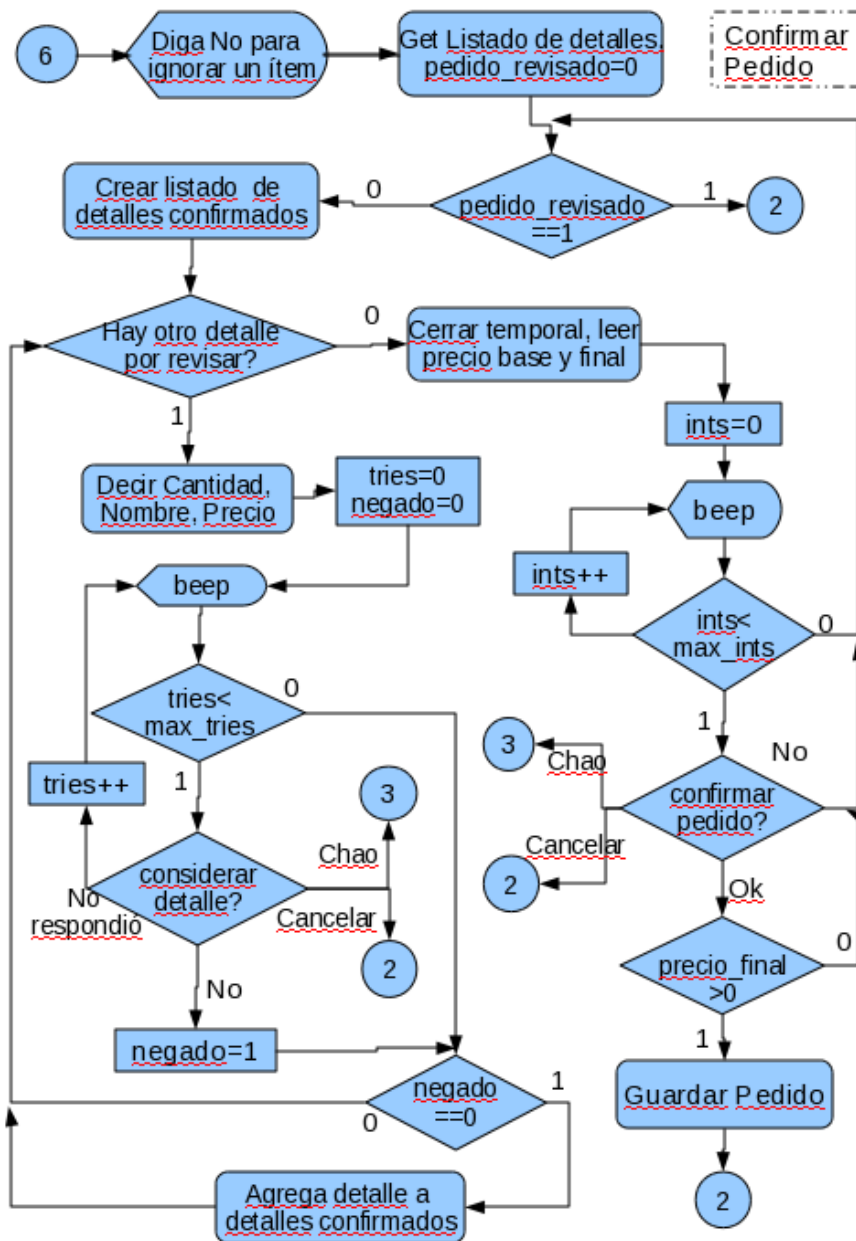


Figura 3.13 Confirmar Pedido.

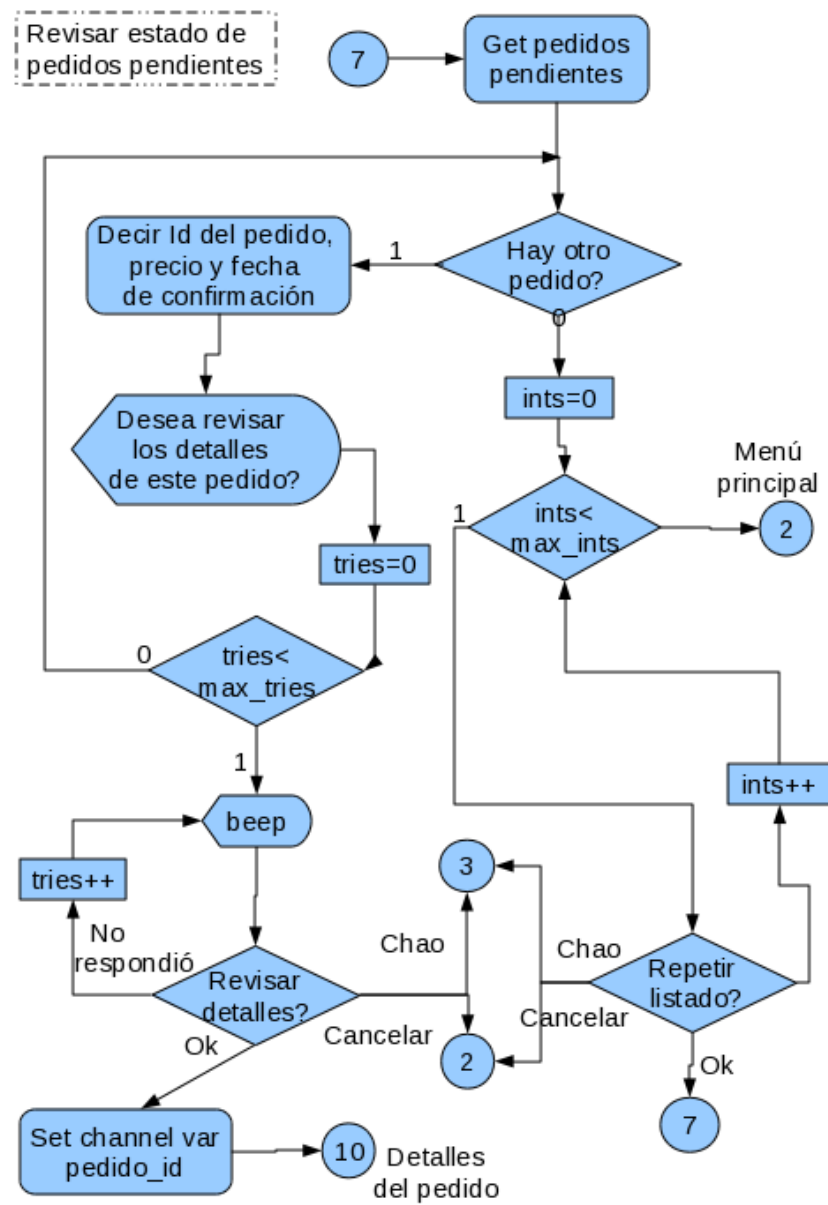


Figura 3.14 Revisar estado de Pedido Pendiente.

3.8.11. Revisar detalles de un pedido

Al ejecutar este módulo se consulta la variable de canal pedido_id. Si ésta no es mayor a cero, se redirecciona al módulo Error. Si la variable es correcta, se verifica que el usuario esté autenticado. Si no está autenticado se le comunica esto y como este caso no es factible (debe haber estado autenticado para oír sus pedidos pendientes), se lo debe redireccionar al módulo Error.

En el caso esperado de que el usuario esté autenticado, se consulta a la base el total de detalles activos correspondientes a este pedido. Si este total es mayor a cero se consultan estos detalles, se recorre este arreglo narrando la cantidad, nombre y precio. Al terminar de listar los detalles del pedido se redirecciona al usuario al listado de pedidos pendientes.

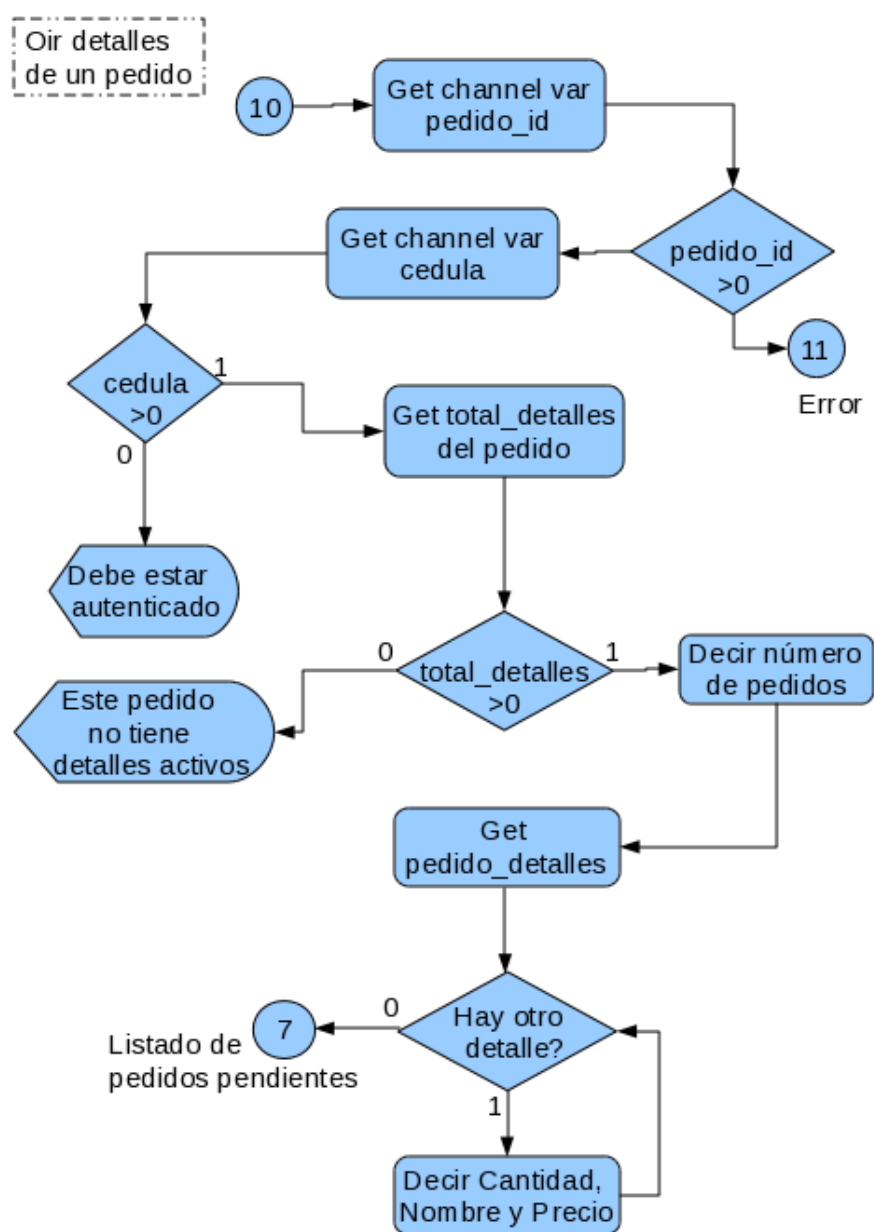


Figura 3.15 Oir detalles de un pedido

3.8.12. Más información

Al llegar a este módulo se le recuerda al usuario que está en Más información. En este módulo se narra toda la información del restaurante dividida en secciones. Entre cada sección se le ofrece al usuario un intento de optar por repetir toda la información. Como en todos los prompts de respuesta de voz, si el usuario dice Chao puede cerrar la llamada.

Luego de leer toda la información se ofrece un máximo de intentos para responder si quiere repetir todas las opciones. Si dice Cancelar el usuario regresa al menú principal y si no contesta repite toda la información. Esto se debería corregir preguntando si quiere salir, y en caso de que no responda, regrese al menú principal.

3.8.13. Chao: Cerrar llamada

En este módulo se elimina el archivo temporal en caso de existir y se le solicita al usuario calificar su nivel de satisfacción con nuestro servicio de IVR ingresando un número del uno al diez por teclado alfanumérico. Esto es opcional y lo puede realizar sin la necesidad de estar autenticado.

Más información

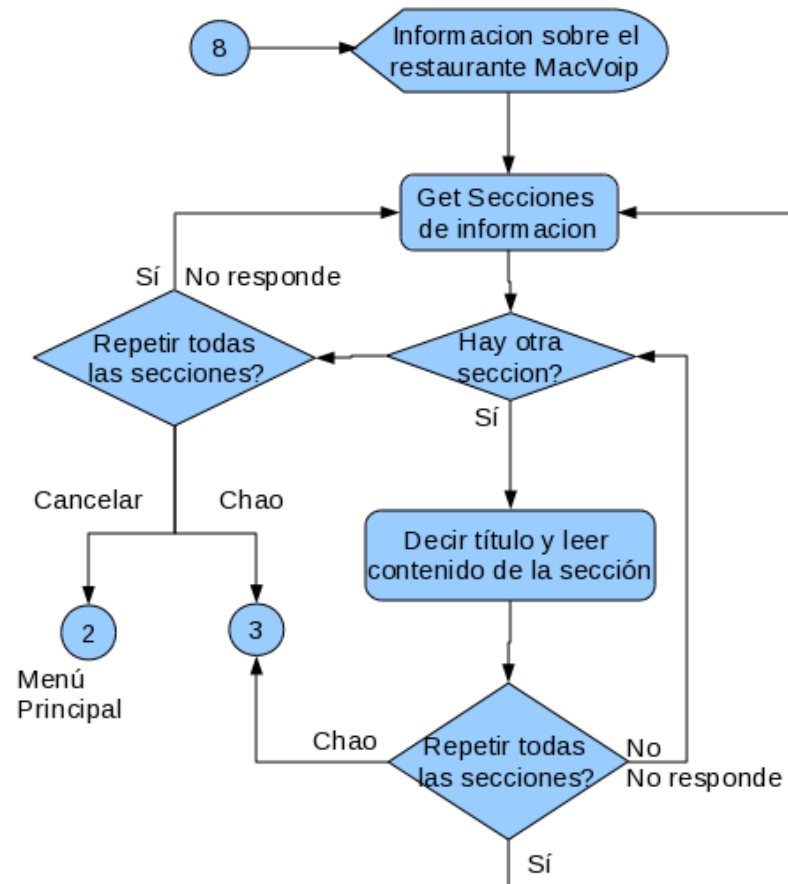


Figura 3.16 Más información

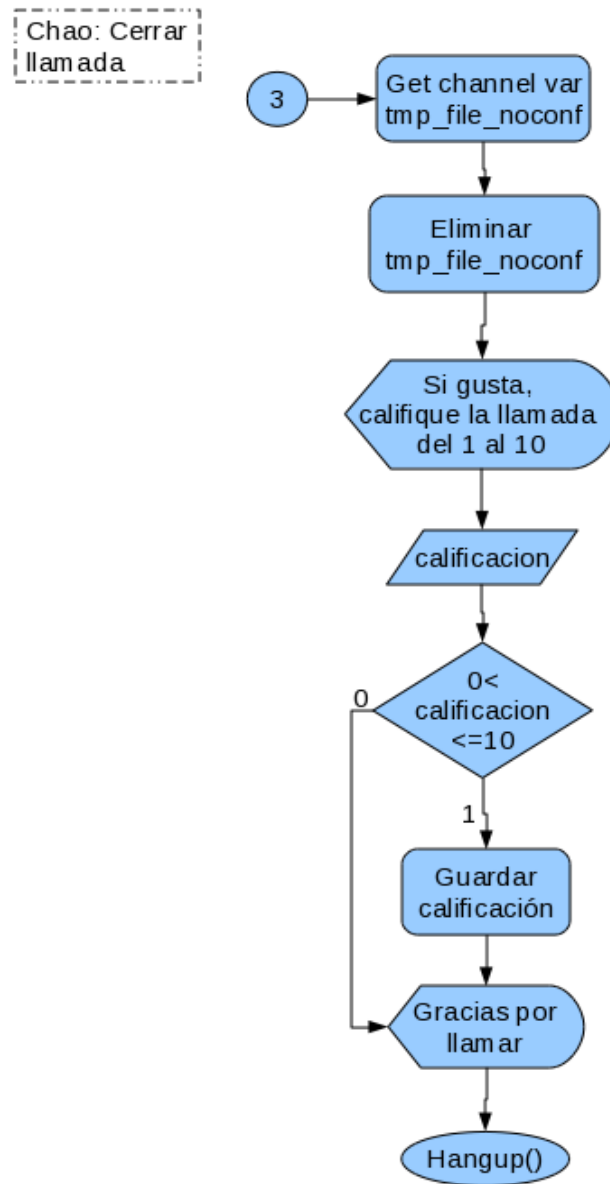


Figura 3.17 Chao: Cerrar llamada.

3.8.14. Error

El objetivo de este módulo es prevenir un corte de la llamada por manejo de datos incorrectos o pérdida de conexión con la base de datos. En el caso de pérdida de conexión de la base de datos se da el problema de que si necesito hacer una consulta o registro, la llamada se corta inmediatamente.

En el caso 'imposible' de que un usuario tenga un ID cero, por ejemplo, si esto está validado debería llegar a este módulo para que el usuario sepa que ha ocurrido algo inesperado y sea redirigido al menú principal.

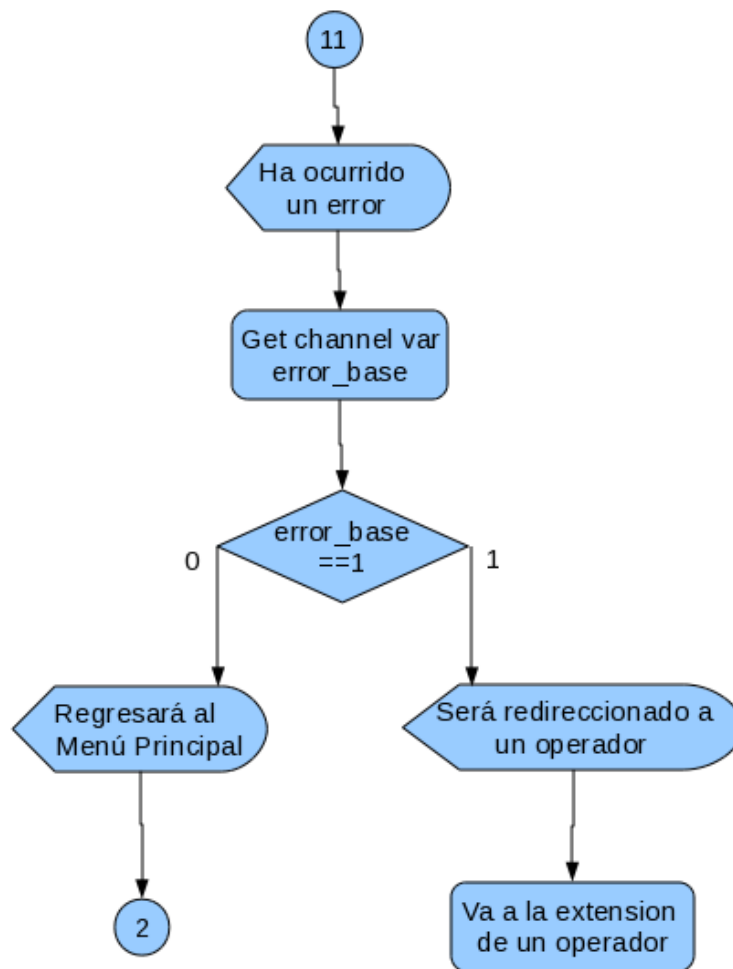


Figura 3.18 Esquema del módulo Error

CAPÍTULO 4

4. PRUEBAS Y RESULTADOS

4.1. Registro de un usuario

La prueba se realizará con un softphone. En éste se crea y registra una cuenta sip que apunta a la IP de nuestro servidor. En este ejemplo se llamará 'cliente'. Se revisa si el usuario 'cliente' está conectado a la central correctamente.

```
[root@localhost bin]# asterisk -rvvv
Verbosity is at least 3
localhost*CLI> sip show peers
Name/username      Host                               Dyn Forcerport ACL Port      Status
cliente/cliente    192.168.1.103                     D  N      5060      OK (12 ms)
contabilidad       (Unspecified)                     D  N        0          UNKNOWN
gerente            (Unspecified)                     D  N        0          UNKNOWN
soporte/soporte    (Unspecified)                     D  N        0          UNKNOWN
4 sip peers [Monitored: 1 online, 3 offline Unmonitored: 0 online, 0 offline]
```

Desde el softphone, 'cliente' llama al número 123, la extensión principal del programa.



Figura 4.1 Llamada al sistema desde el softphone

Desde la consola de Asterisk en el servidor podemos ver que se recibe la llamada de 'cliente' y se ejecuta el primer AGI, bienvenida. Aquí, al comprobar que no hay una variable de canal 'cedula', ejecuta el módulo 'Autenticar cliente'. Al llegar se pide al usuario que ingrese un número de cédula después del tono.

```
localhost*CLI>
== Using SIP RTP CoS mark 5
-- Executing [123@asr:1] Answer("SIP/cliente-00000004", "") in new stack
-- Executing [123@asr:2] AGI("SIP/cliente-00000004", "bienvenida.agi") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/bienvenida.agi
-- Playing 'tts/tts-3ad9a591f661bbe519ec0f57d43502cc' (escape_digits=) (sample_offset 0)
bienvenida.agi: En bienvenida, cedula:
bienvenida.agi: Va a autenticar cliente.
-- AGI Script Executing Application: (AGI) Options: (autenticar_cliente.agi)
-- Launched AGI Script /var/lib/asterisk/agi-bin/autenticar_cliente.agi
-- Playing 'tts/tts-9ed531725d5ed555511ffd2cf11b4c51' (escape_digits=) (sample_offset 0)
-- <SIP/cliente-00000004> Playing 'beep.gsm' (language 'es')
localhost*CLI> █
```

Figura 4.2 CLI de Asterisk: Recepción de la llamada



Figura 4.3 Ingreso del número de cédula en el softphone

Se registrará a un usuario con número de cédula 0987987987 y contraseña 987987.

```

-- Playing 'tts/tts-b0f1e8cd8d93a9fb025f1bfda82aae1d' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: En decirCedula
-- Playing 'tts/tts-cfcd208495d565ef66e7dff9f98764da' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: numero: 0
-- Playing 'tts/tts-45c48cce2e2d7fbdea1afc51c7c6ad26' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: numero: 9
-- Playing 'tts/tts-c9f0f895fb98ab9159f51fd0297e236d' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: numero: 8
-- Playing 'tts/tts-8f14e45fceeal67a5a36dedd4bea2543' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: numero: 7
-- Playing 'tts/tts-45c48cce2e2d7fbdea1afc51c7c6ad26' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: numero: 9
-- Playing 'tts/tts-c9f0f895fb98ab9159f51fd0297e236d' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: numero: 8
-- Playing 'tts/tts-8f14e45fceeal67a5a36dedd4bea2543' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: numero: 7
-- Playing 'tts/tts-45c48cce2e2d7fbdea1afc51c7c6ad26' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: numero: 9
-- Playing 'tts/tts-c9f0f895fb98ab9159f51fd0297e236d' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: numero: 8
-- Playing 'tts/tts-8f14e45fceeal67a5a36dedd4bea2543' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: numero: 7
autenticar_cliente.agi: cedula: 0987987987, cedula_registrada: 0
-- Playing 'tts/tts-a9110e9149883536170a84fe92df6c6e' (escape_digits=) (sample_offset 0)
-- Playing 'beep' (escape_digits=) (sample_offset 0)
localhost*CLI> █

```

Figura 4.4 CLI de Asterisk: Lectura de número de cédula

Se revisa si el número de cédula ingresado está registrado en la base o no. En este caso el número es nuevo así que se creará un nuevo usuario. Se le pregunta al usuario si desea registrarse. Esto le da una oportunidad de ingresar otro número de cédula si ingresó uno incorrecto.

```

autenticar_cliente.agi: cedula: 0987987987, cedula_registrada: 0
  -- Playing 'tts/tts-a9110e9149883536170a84fe92df6c6e' (escape_digits=) (sample_offset 0)
localhost*CLI>
  -- Playing 'beep' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: # Intento responderDeseaRegistrarse: 0, vresponse: OHKAY
  -- Playing 'tts/tts-08be8f1c8956d12f56f925e42dcd37b3' (escape_digits=) (sample_offset 0)
  -- <SIP/cliente-00000007> Playing 'beep.gsm' (language 'es')
  -- Playing 'tts/tts-f20516b251c1f443021558c987389f94' (escape_digits=) (sample_offset 0)
  -- <SIP/cliente-00000007> Playing 'beep.gsm' (language 'es')
autenticar_cliente.agi: Contrasenía confirmada correctamente
  -- Playing 'tts/tts-be7854dc15915ff7cdb449dd1f9b2ef1' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: En registrarCedulaPersona, query: insert into Persona(Fecha_Registro,
Password, Persona_Tipo_id, Cedula) values(NOW(), '987987', 1, '0987987987');
  -- Playing 'tts/tts-80d608de8b21ebb77271e486f49db88c' (escape_digits=) (sample_offset 0)
autenticar_cliente.agi: Usuario registrado exitosamente
autenticar_cliente.agi: Va al Menu principal
  -- AGI Script Executing Application: (AGI) Options: (menu_principal.agi)
  -- Launched AGI Script /var/lib/asterisk/agi-bin/menu_principal.agi
  -- Playing 'tts/tts-b803620528782ce62fb3c2fc880014a2' (escape_digits=) (sample_offset 0)
menu_principal.agi: Luego en Menu principal, cedula: 0987987987
menu_principal.agi: num_pedidos_noterminados: 0
menu_principal.agi: Opciones Menu principal:
menu_principal.agi: opcion_id: 0, nombre_opcion: Menu de categoriias
  -- Playing 'tts/tts-0047f7a6db53ef8e3b19de8662f2e3ca' (escape_digits=) (sample_offset 0)
  -- <SIP/cliente-00000007>AGI Script menu_principal.agi completed, returning -1
  -- <SIP/cliente-00000007>AGI Script autenticar_cliente.agi completed, returning 0
  -- <SIP/cliente-00000007>AGI Script bienvenida.agi completed, returning 0

```

Figura 4.5 CLI de Asterisk: Ingreso y confirmación de contraseña

Desde 'cliente' se respondió Ok. Se solicita que ingrese una contraseña de entre 4 y 8 dígitos y se la confirme.



Figura 4.6 Ingreso y confirmación de contraseña desde el softphone

Si esto se realiza correctamente, en consola podemos ver el query que se hace a la base. Si el query logra registrar al usuario correctamente, se muestra esto en consola. Luego de registrar al usuario se ejecuta el módulo del Menú Principal.

4.2. Selección de un pedido

En el Menú Principal se revisa si hay una cédula autenticada y si tiene pedidos pendientes para construir el menú. En el menú principal se narra las opciones y en este ejemplo 'cliente' va directamente al Menú de Categorías diciendo Ok. Al llegar a este menú se consulta el número de categorías activas a listarse y se lo muestra en consola.

```
localhost*CLI>
menu_principal.agi: Luego en Menu principal, cedula: 0987987987
menu_principal.agi: num_pedidos_noterminados: 0
menu_principal.agi: Opciones Menu principal:
menu_principal.agi: opcion_id: 0, nombre_opcion: Menu de categoriias
  -- Playing 'tts/tts-0047f7a6db53ef8e3b19de8662f2e3ca' (escape_digits=) (sample_offset 0)
  -- Playing 'beep' (escape_digits=) (sample_offset 0)
menu_principal.agi: # Intento elegirOpcion: 0, vresponse: OKAY
menu_principal.agi: opcion_id: 0
menu_principal.agi: Va al Menu de Categorias.
  -- AGI Script Executing Application: (AGI) Options: (select_categoria.agi)
  -- Launched AGI Script /var/lib/asterisk/agi-bin/select_categoria.agi
select_categoria.agi: Llega a select_categoria
select_categoria.agi: Cuantas cats: 4
  -- Playing 'tts/tts-5369a66a995bbec5696d5b2ac4d141b3' (escape_digits=) (sample_offset 0)
localhost*CLI> □
```

Figura 4.7 CLI de Asterisk: Selección en menú principal

Se selecciona el Menú de categorías y dentro, se espera a la opción Postre para decir Ok. Revisar figura 4.8.

Como se ve en la figura 4.9, En el menú de Postre, se ha seleccionado dos flanes y luego se regresa al de menú de categorías presionando el número 99.

```

menu_principal.agi: # Intento elegirOpcion: 0, vresponse:
-- Playing 'beep' (escape_digits=) (sample_offset 0)
menu_principal.agi: # Intento elegirOpcion: 1, vresponse: OHKAY
menu_principal.agi: opcion_id: 0
menu_principal.agi: Va al Menu de Categorias.
-- AGI Script Executing Application: (AGI) Options: (select_categoria.agi)
-- Launched AGI Script /var/lib/asterisk/agi-bin/select_categoria.agi
select_categoria.agi: Llega a select_categoria
select_categoria.agi: Cuantas cats: 4
-- Playing 'tts/tts-5369a66a995bbbee5696d5b2ac4d141b3' (escape_digits=) (sample_offset 0)
-- Playing 'tts/tts-3aad6cc63303546f16515bbf4b46da2' (escape_digits=) (sample_offset 0)
-- Playing 'tts/tts-cdf4d38e2deblee57b73a8d8120b5fdc' (escape_digits=) (sample_offset 0)
-- Playing 'tts/tts-89690b1e99a8a3a60c962e01933a1773' (escape_digits=) (sample_offset 0)
-- Playing 'beep' (escape_digits=) (sample_offset 0)
select_categoria.agi: # Intento elegirCategoria: 0, vresponse:
-- Playing 'beep' (escape_digits=) (sample_offset 0)
select_categoria.agi: # Intento elegirCategoria: 1, vresponse:
-- Playing 'tts/tts-7810ba92c1233a00f5d90f548ddc44f0' (escape_digits=) (sample_offset 0)
-- Playing 'beep' (escape_digits=) (sample_offset 0)
select_categoria.agi: # Intento elegirCategoria: 0, vresponse:
-- Playing 'beep' (escape_digits=) (sample_offset 0)
select_categoria.agi: # Intento elegirCategoria: 1, vresponse:
-- Playing 'tts/tts-3734afe85d229b971d0a016d8717fcea' (escape_digits=) (sample_offset 0)
-- Playing 'beep' (escape_digits=) (sample_offset 0)
select_categoria.agi: # Intento elegirCategoria: 0, vresponse: OKAY
-- AGI Script Executing Application: (AGI) Options: (select_item.agi)
-- Launched AGI Script /var/lib/asterisk/agi-bin/select_item.agi
select_item.agi: Cédula autenticada, presentarItemsParaSeleccionar
-- <SIP/cliente-0000000a> Playing 'tts/tts-8d360a49ca79e95a17cb50b4cbb1209e.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-a9203548209b49f7589dfa760f867a69.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-9bf31c7ff062936a96d3c8bd1f8f2ff3.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-7cc3789ca6f4e20f2e2b9c5dffb0b8be9.slin' (language 'es')
localhost*CLI>
-- <SIP/cliente-0000000a> Playing 'tts/tts-8c2dee7ad88c1c36062ce9841b98586e.slin' (language 'es')

```

Figura 4.8 CLI de Asterisk: Menú de Categorías

```

-- <SIP/cliente-0000000a> Playing 'tts/tts-4885a3c2b708cccefc0e4bbfdf68e071.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-02f50d6456c23d34eb40b612d95ab025.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'beep.gsm' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-462c60395f0e4503d93f994204dc9ef9.slin' (language 'es')
select_item.agi: texto_linea: 11,2
-- <SIP/cliente-0000000a> Playing 'tts/tts-4f8c2c8ae8f34297e0a92224ece9b4df.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'beep.gsm' (language 'es')
select_item.agi: Presiono el codigo para regresar al menu de categorias
-- AGI Script Executing Application: (AGI) Options: (select_categoria.agi)
-- Launched AGI Script /var/lib/asterisk/agi-bin/select_categoria.agi
select_categoria.agi: Llega a select_categoria
select_categoria.agi: Cuantas cats: 4
-- Playing 'tts/tts-5369a66a995bbbee5696d5b2ac4d141b3' (escape_digits=) (sample_offset 0)

```

Figura 4.9 CLI de Asterisk: Selección de ítems

Luego se selecciona la primera categoría, Comida, diciendo Ok, lo cual es reconocido al segundo intento. Se elige dos hamburguesas vegetarianas y un taco y luego se presiona el código 98 para revisar el pedido actual.

```

-- Playing 'tts/tts-3aadb6cc63303546f16515bbf4b46da2' (escape_digits=) (sample_offset 0)
-- Playing 'tts/tts-cdf4d38e2deb1ee57b73a8d8120b5fdc' (escape_digits=) (sample_offset 0)
-- Playing 'tts/tts-89690b1e99a8a3a60c962e01933a1773' (escape_digits=) (sample_offset 0)
-- Playing 'beep' (escape_digits=) (sample_offset 0)
select_categoria.agi: # Intento elegirCategoria: 0, vresponse:
-- Playing 'beep' (escape_digits=) (sample_offset 0)
select_categoria.agi: # Intento elegirCategoria: 1, vresponse: OHKAY
-- AGI Script Executing Application: (AGI) Options: (select_item.agi)
-- Launched AGI Script /var/lib/asterisk/agi-bin/select_item.agi
select_item.agi: Cédula autenticada, presentarItemsParaSeleccionar
-- <SIP/cliente-0000000a> Playing 'tts/tts-8d360a49ca79e95a17cb50b4cbb1209e.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-a9203548209b49f7589dfa760f867a69.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-9bf31c7ff062936a96d3c8bd1f8f2ff3.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-7cc3789ca6f4e20f2e2b9c5dfb0b8be9.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-8c2dee7ad88c1c36062ce9841b98586e.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-4885a3c2b708cccefc0e4bbfdf68e071.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-41061c2a0fb9bb67cd6e91230a7fa215.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'beep.gsm' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-05026c8ad6accf77bf2aa463b6faf48c.slin' (language 'es')
select_item.agi: texto_linea: 1,2
-- <SIP/cliente-0000000a> Playing 'tts/tts-972e226430e421c35d16961b0477b4ec.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'beep.gsm' (language 'es')
-- <SIP/cliente-0000000a> Playing 'tts/tts-555b9acab01c5024e15bb96f07b1927d.slin' (language 'es')
select_item.agi: texto_linea: 2,1
-- <SIP/cliente-0000000a> Playing 'tts/tts-9d3aaaefe631b340159f29143b6a8d15d.slin' (language 'es')
-- <SIP/cliente-0000000a> Playing 'beep.gsm' (language 'es')
select_item.agi: tmp_file_noconf: data_noconf.txt
select_item.agi: Va a Revisar Pedido.
-- AGI Script Executing Application: (AGI) Options: (confirmar_pedido.agi)
-- Launched AGI Script /var/lib/asterisk/agi-bin/confirmar_pedido.agi
confirmar_pedido.agi: En Confirmar Pedido, cedula: 0987987987, tmp_file_noconf: data_noconf.txt, length_cedula: 10
localhost*CLI>
-- Playing 'tts/tts-9625a9cc8017f6fa4b9d79699111c055' (escape_digits=) (sample_offset 0)

```

Figura 4.10 CLI de Asterisk: Selección de ítems. Selecciona confirmar pedido

En la penúltima línea de la figura 4.10. vemos que al llegar al AGI confirmar_pedido se revisa la cédula y si existe un archivo temporal con los ítems seleccionados. En el módulo de Confirmar Pedido se listan en este orden: dos hamburguesas vegetarianas, dos flanes y un taco. Al oír el detalle de '1 Taco' desde 'cliente' se

dice No para ignorar este detalle. La última línea es la ejecución de la respuesta 'Se ignorará este detalle en su pedido'.

```

confirmar_pedido.agi: item_id: 1, cantidad: 2, detalle_precio: 6.6
  -- Playing 'beep' (escape_digits=) (sample_offset 0)
confirmar_pedido.agi: # Intento negar detalle: 0, ultima: OKAY
  -- Playing 'beep' (escape_digits=) (sample_offset 0)
confirmar_pedido.agi: # Intento negar detalle: 1, ultima: ZEE
confirmar_pedido.agi: Agregar al hash el item_id 1, 2 unidades
confirmar_pedido.agi: No existia el item_id 1
  -- Playing 'tts/tts-d40498d8c62e0f010835c86fb505fall' (escape_digits=) (sample_offset 0)
confirmar_pedido.agi: item_id: 11, cantidad: 2, detalle_precio: 0.8
  -- Playing 'beep' (escape_digits=) (sample_offset 0)
confirmar_pedido.agi: # Intento negar detalle: 0, ultima: ZEE
  -- Playing 'beep' (escape_digits=) (sample_offset 0)
confirmar_pedido.agi: # Intento negar detalle: 1, ultima: ZEE
confirmar_pedido.agi: Agregar al hash el item_id 11, 2 unidades
confirmar_pedido.agi: No existia el item_id 11
  -- Playing 'tts/tts-0d2f9aec690c9457e424c87728c68ec9' (escape_digits=) (sample_offset 0)
confirmar_pedido.agi: item_id: 2, cantidad: 1, detalle_precio: 2
  -- Playing 'beep' (escape_digits=) (sample_offset 0)
confirmar_pedido.agi: # Intento negar detalle: 0, ultima: ZEE
  -- Playing 'beep' (escape_digits=) (sample_offset 0)
confirmar_pedido.agi: # Intento negar detalle: 1, ultima: NOH
  -- Playing 'tts/tts-86fb878ee1a226a747ee5ad8d7692c9f' (escape_digits=) (sample_offset 0)

```

Figura 4.11 CLI de Asterisk: Confirmar pedido

Como se ve en la figura 4.12, luego de listar los ítems se dice el precio base y precio final (sumado el IVA) del pedido considerando los detalles no anulados. Se pregunta si se desea confirmar el pedido. En este caso 'cliente' dice Ok.

```

confirmar_pedido.agi: precio_base_pedido: 7.4, num_items_confirmados: buscar
confirmar_pedido.agi: precio_final_pedido: 8.288
  -- Playing 'tts/tts-e217f2a2877409b58e54f3b807917d14' (escape_digits=) (sample_offset 0)
  -- Playing 'tts/tts-77e302bc29d0239dcade011086d5872a' (escape_digits=) (sample_offset 0)
  -- Playing 'tts/tts-7b5c011f2fd42869529056fb9fbd6fc3' (escape_digits=) (sample_offset 0)
  -- Playing 'tts/tts-dad8622f32edcbcllee505546ea54e7b5' (escape_digits=) (sample_offset 0)
  -- Playing 'tts/tts-1abdab173c6b000588842f79182e2e53' (escape_digits=) (sample_offset 0)
  -- Playing 'tts/tts-elid6488412118a5d0674153e1a134081' (escape_digits=) (sample_offset 0)
  -- Playing 'beep' (escape_digits=) (sample_offset 0)
confirmar_pedido.agi: #Intento confirmar pedido: 0, ultima palabra: OKAY
confirmar_pedido.agi: Va a guardar el pedido.

```

Figura 4.12 CLI de Asterisk: Guardar pedido

En la figura 4.13 se ve que en consola se muestra el ID del pedido ingresado y cuáles fueron los detalles guardados. También se confirma que se ha borrado el archivo temporal y su variable de canal. Se ejecuta el módulo Menú Principal.

```

confirmar_pedido.agi: pedido_id: 5
confirmar_pedido.agi: items_confirmados:
confirmar_pedido.agi: item_id: 1, cantidad: 2
confirmar_pedido.agi: item_id: 11, cantidad: 2
confirmar_pedido.agi: file_deleted: 1
confirmar_pedido.agi: Se blanqueo variable del tmp
  -- Playing 'tts/tts-2284df88b63539a628f56305a14c2a63' (escape_digits=) (sample_offset 0)
  -- Playing 'tts/tts-80d6cd8e79a780f514ad9d21d8adf3e0' (escape_digits=) (sample_offset 0)
confirmar_pedido.agi: Va al Menu principal
  -- AGI Script Executing Application: (AGI) Options: (menu_principal.agi)

```

Figura 4.13 CLI de Asterisk: Se registró el pedido, va a menú principal

4.3. Revisión de un pedido

En el menú principal el usuario selecciona la opción Revisar estado del pedido pendiente. Se ejecuta el AGI `revisar_estado_pedidos`.

```

menu_principal.agi: # Intento elegirOpcion: 1, vresponse:
menu_principal.agi: opcion_id: 2, nombre_opcion: Revisar estado del pedido pendiente
localhost*CLI>
  -- Playing 'tts/tts-b05ab9e3be07f6ac7b9c74cf73ddc3a4' (escape_digits=) (sample_offset 0)
localhost*CLI>
  -- Playing 'beep' (escape_digits=) (sample_offset 0)
menu_principal.agi: # Intento elegirOpcion: 0, vresponse: ZEE OHKAY
menu_principal.agi: opcion_id: 2
menu_principal.agi: Va a Revisar estado del pedido.
  -- AGI Script Executing Application: (AGI) Options: (revisar_estado_pedidos.agi)
  -- Launched AGI Script /var/lib/asterisk/agi-bin/revisar_estado_pedidos.agi
  -- Playing 'tts/tts-04762ec39f76d9c82724a8e2913d83dd' (escape_digits=) (sample_offset 0)
revisar_estado_pedidos.agi: En revisar_estado_pedidos, cedula: 0987987987, va a presentarResumenPedidos
revisar_estado_pedidos.agi: pedido[0]: 5, pedido[1]: 8.29, pedido[2]: Confirmado, pedido[3]: 2011-08-29 05:36:53
  -- Playing 'tts/tts-48281cf858244e766f53a3cfbbe19db' (escape_digits=) (sample_offset 0)
revisar_estado_pedidos.agi: # Intento accederADetallesPedido: 0, vresponse: OHKAY
revisar_estado_pedidos.agi: Va a revisar los detalles del pedido 5
  -- AGI Script Executing Application: (AGI) Options: (detalles_pedido.agi)
  -- Launched AGI Script /var/lib/asterisk/agi-bin/detalles_pedido.agi

```

Figura 4.14 CLI de Asterisk: Revisar estado de pedidos pendientes

Al llegar a Detalles del pedido seleccionado se muestra en consola el número de detalles que tiene ese pedido y narra cada detalle.

```

detalles_pedido.agi: El pedido 5 tiene 2 detalles.
detalles_pedido.agi: Cédula autenticada, listar detalles del pedido
  -- <SIP/cliente-00000003> Playing 'tts/tts-6b14aeaff7e34a54b7af96c13a4079ac.slin' (language 'es')
detalles_pedido.agi: row[5]:7.39, row[1]: 2, row[0]: Hamburguesa vegetariana
  -- <SIP/cliente-00000003> Playing 'tts/tts-e988aae41ee937ea598726dcd6ef26.slin' (language 'es')
detalles_pedido.agi: row[5]:0.90, row[1]: 2, row[0]: Flan
  -- <SIP/cliente-00000003> Playing 'tts/tts-1faf3be23b83da866c1ab2dee9e57de5.slin' (language 'es')
  -- <SIP/cliente-00000003> Playing 'tts/tts-ef8739424204438bac09d68cd71410ec.slin' (language 'es')
detalles_pedido.agi: Va a Revisar estado del pedido.
  -- AGI Script Executing Application: (AGI) Options: (revisar_estado_pedidos.agi)
  -- Launched AGI Script /var/lib/asterisk/agi-bin/revisar_estado_pedidos.agi

```

Figura 4.15 CLI de Asterisk: Revisar detalles de pedido seleccionado

Al terminar de listar los detalles se regresa al listado de pedidos pendientes. Aquí, 'cliente' decide terminar la llamada diciendo Chao. Se ejecuta el módulo Chao: Cerrar llamada.

```
-- Playing 'tts/tts-04762ec39f76d9c82724a8e2913d83dd' (escape_digits=) (sample_offset 0)
revisar_estado_pedidos.agi: En revisar_estado_pedidos, cedula: 0987987987, va a presentarResumenPedidos
revisar_estado_pedidos.agi: pedido[0]: 5, pedido[1]: 8.29, pedido[2]: Confirmado, pedido[3]: 2011-08-29 05:36:53
-- Playing 'tts/tts-48281cf858244e766f53a3cfbbee19db' (escape_digits=) (sample_offset 0)
-- Playing 'tts/tts-ba0c666f7328fd8b79c93b480d05ae92' (escape_digits=) (sample_offset 0)
-- Playing 'tts/tts-df6274b64c5b63206bcef983747804df' (escape_digits=) (sample_offset 0)
-- Playing 'tts/tts-ca5dd64f0f32317e720752e691549c66' (escape_digits=) (sample_offset 0)
-- Playing 'tts/tts-0d6f228cbe906db32e61ff098757303c' (escape_digits=) (sample_offset 0)
-- Playing 'beep' (escape_digits=) (sample_offset 0)
revisar_estado_pedidos.agi: # Intento accederADetallesPedido: 0, vresponse: CHAO ZEE
-- AGI Script Executing Application: (AGI) Options: (chao.agi)
-- Launched AGI Script /var/lib/asterisk/agi-bin/chao.agi
```

Figura 4.16 CLI de Asterisk: Listado de detalles del pedido

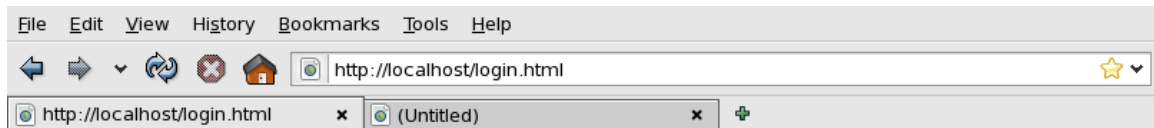
Aquí se solicita al usuario una calificación a la llamada, donde ingresó 9. Se registra este voto y se cierra la llamada.

```
chao.agi: No habia tmp_file_noconf
-- <SIP/cliente-00000003> Playing 'tts/tts-7461076c5875d3870f6a32da33ce86f8.slin' (language 'es')
-- <SIP/cliente-00000003> Playing 'beep.gsm' (language 'es')
chao.agi: En guardarCalificacionLlamada, calificacion: 9, cedula: 0987987987, callerid: cliente, uniqueid: 1314615120.3
-- <SIP/cliente-00000003> Playing 'tts/tts-121fd0a331079f3360791402adc95f0e.slin' (language 'es')
chao.agi: Gracias por darnos su opinion. Hasta pronto. Hangup.
-- <SIP/cliente-00000003>AGI Script chao.agi completed, returning -1
-- <SIP/cliente-00000003>AGI Script revisar_estado_pedidos.agi completed, returning 0
-- <SIP/cliente-00000003>AGI Script detalles_pedido.agi completed, returning -1
-- <SIP/cliente-00000003>AGI Script revisar_estado_pedidos.agi completed, returning -1
-- <SIP/cliente-00000003>AGI Script menu_principal.agi completed, returning -1
-- <SIP/cliente-00000003>AGI Script autenticar_cliente.agi completed, returning -1
-- <SIP/cliente-00000003>AGI Script bienvenida.agi completed, returning -1
```

Figura 4.17 CLI de Asterisk: Se cierra la llamada

4.4. Inicio de sesión en la aplicación interna de empleados

Pantalla de Inicio de Sesión para acceder a la aplicación de los empleados del restaurante.



Sistema de Toma de Pedidos

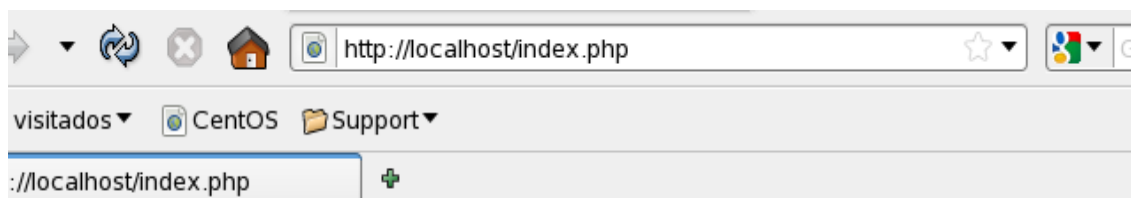
Ingrese su Cedula y Contraseña

Cedula :	<input type="text" value="0925446106"/>
Contraseña :	<input type="password" value="....."/>
<input type="button" value="Ingresar"/>	

Figura 4.18 Pantalla de Inicio de Sesión de aplicación interna

4.5. Menú principal de la aplicación web

Esta es la pantalla principal de la aplicación, donde se muestran los reportes disponibles para los empleados del restaurante.



Sistema de Toma de Pedidos

Bienvenido Alvaro Padilla

[Consultar Pedidos Pendientes](#)

[Consultar Pedidos Pagados](#)

[Consultar Clientes](#)

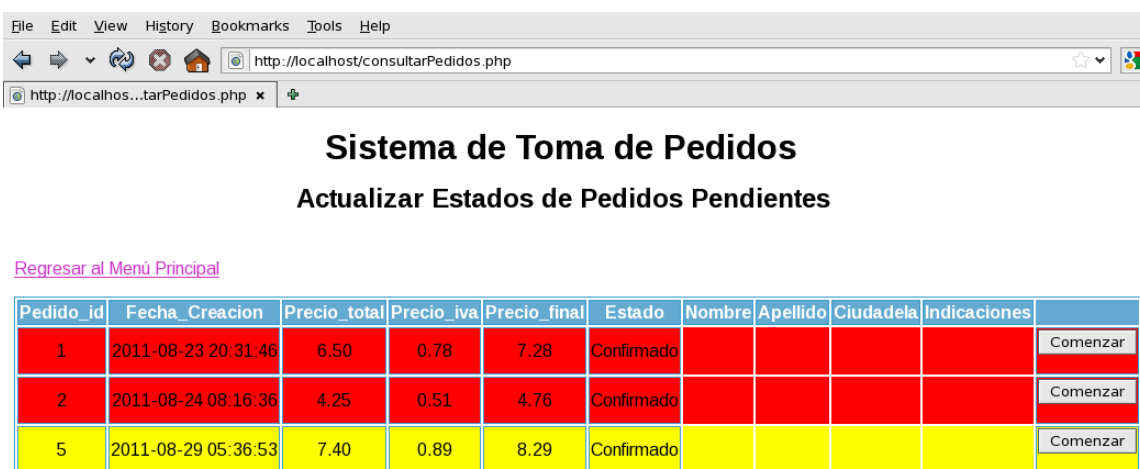
[Salir del Sistema](#)

Figura 4.19 Pantalla principal de aplicación interna

4.6. Reporte principal: Estados de pedidos pendientes

Se muestra todos los pedidos no anulados pendientes de pago y se da la opción de actualizar su estado con el botón a la derecha.

El pedido creado en este ejemplo está amarillo. Al darle click se accede a sus detalles.



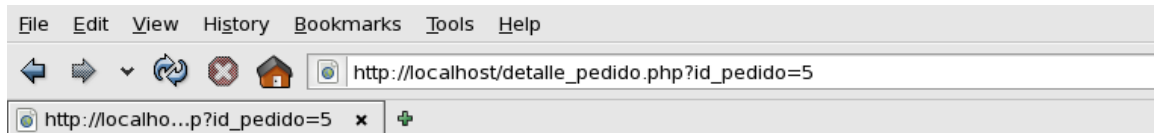
The screenshot shows a web browser window with the URL `http://localhost/consultarPedidos.php`. The page title is "Sistema de Toma de Pedidos" and the subtitle is "Actualizar Estados de Pedidos Pendientes". There is a link "Regresar al Menú Principal" and a table of orders. The table has 11 columns: Pedido_id, Fecha_Creacion, Precio_total, Precio_iva, Precio_final, Estado, Nombre, Apellido, Ciudadela, Indicaciones, and a 'Comenzar' button. The row with Pedido_id 5 is highlighted in yellow.

Pedido_id	Fecha_Creacion	Precio_total	Precio_iva	Precio_final	Estado	Nombre	Apellido	Ciudadela	Indicaciones	Comenzar
1	2011-08-23 20:31:46	6.50	0.78	7.28	Confirmado					Comenzar
2	2011-08-24 08:16:36	4.25	0.51	4.76	Confirmado					Comenzar
5	2011-08-29 05:36:53	7.40	0.89	8.29	Confirmado					Comenzar

Figura 4.20 Actualizar estado de pedidos pendientes

4.7. Detalles del pedido seleccionado

Aquí se ve los detalles que se consideraron en el pedido y que serán facturados.



Sistema de Toma de Pedidos

Detalle del Pedido

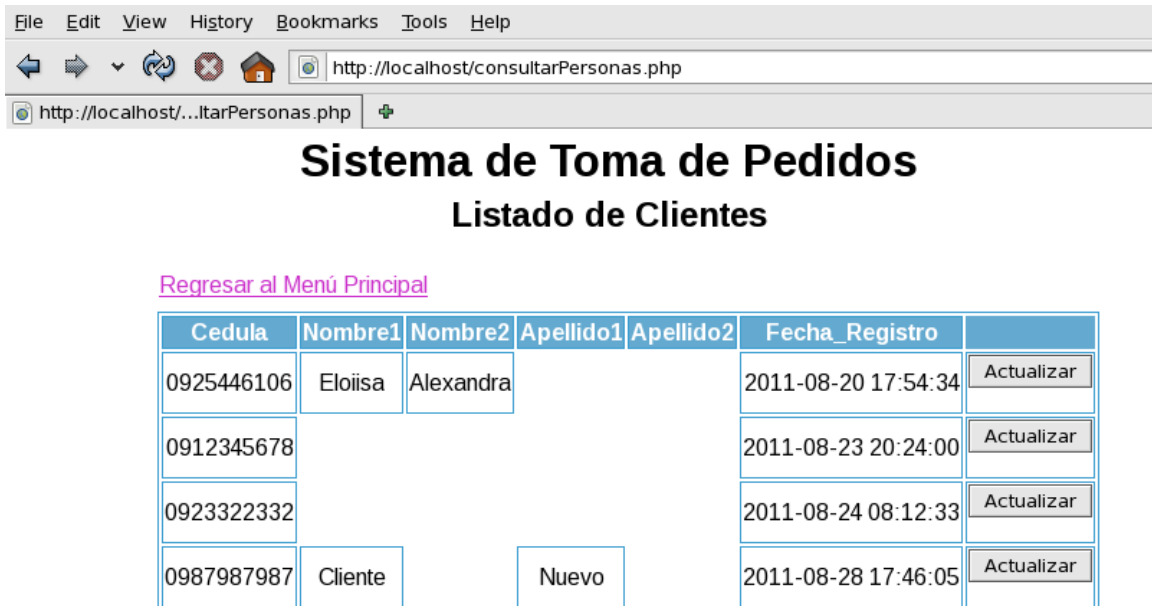
[Regresar](#) [Consultar Pedidos](#)

Nombre	Activo	Unidades	Precio_unidad	Precio_iva	Precio_final
Hamburguesa vegetariana	1	2	3.30	0.79	7.39
Flan	1	2	0.40	0.10	0.90

Figura 4.21 Detalles del Pedido

4.8. Listado de clientes existentes

En esta sección se muestra el listado de todos los clientes registrados en el sistema con la opción a actualizar sus datos.



Sistema de Toma de Pedidos
Listado de Clientes

[Regresar al Menú Principal](#)

Cedula	Nombre1	Nombre2	Apellido1	Apellido2	Fecha_Registro	Actualizar
0925446106	Eloisa	Alexandra			2011-08-20 17:54:34	Actualizar
0912345678					2011-08-23 20:24:00	Actualizar
0923322332					2011-08-24 08:12:33	Actualizar
0987987987	Cliente			Nuevo	2011-08-28 17:46:05	Actualizar

Figura 4.22 Listado de clientes

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. Al ofrecer este sistema de respuesta interactiva, el restaurante McVoip tendría una ventaja frente a su competencia por la innovación con el uso de la tecnología de reconocimiento de voz que no es popular en nuestro medio.
2. En este sistema se logró reconocer vocabulario del usuario en un canal de llamada a Asterisk y utilizar estos como comandos para dirigir el flujo de la llamada recorriendo las opciones y ejecutando acciones de consulta y registro a la base de datos.
3. Se utilizó el procedimiento contrario para comunicarse con el cliente, la conversión de texto a voz. Esto se logró utilizando el sistema Festival para 'leer' los mensajes que la aplicación genera para el usuario. El empleo de Festival nos ha permitido armar los mensajes que escuchará el usuario a partir de variables del canal o información obtenida de la base de datos.
4. Se ha buscado personalizar la aplicación en lo posible tratando de guiar al usuario con mensajes concisos sobre lo que debe hacer y explicando

claramente los resultados de las acciones que ha realizado y la ubicación en la que se encuentra en el flujo de la llamada.

5. Se ha buscado cumplir con el objetivo de optimizar recursos con esta aplicación para entregar esta retroalimentación al usuario utilizando una voz artificial en lugar de mensajes pregrabados de voz. La segunda opción implicaría que una misma persona grabe nuevos nombres de productos, indicaciones, precios, mensajes de error y mensajes de retroalimentación en caso de hacer una actualización.
6. Una limitación de la presente aplicación es que no se puede garantizar que esté disponible para un gran número de usuarios ni ofrecer estimaciones al respecto. De Asterisk existe una recopilación [23] de especificaciones de hardware y testimonios de usuarios que permite estimar el número de llamadas concurrentes y las dimensiones que puede alcanzar nuestra solución.
7. No encontramos datos objetivos similares sobre la capacidad de atención de requerimientos concurrentes y fidelidad para el servicio de conversión de texto a voz de Festival ni del servicio de reconocimiento de voz del Sphinx2, sino solo opiniones de que la calidad baja drásticamente. Se asume

entonces que estos dos servicios no rendirían en un ambiente comercial.

8. No se pudo alcanzar un entendimiento del lenguaje natural porque para esto debíamos contar con un modelo acústico en español y utilizarlo para reconocer un diccionario extenso de posibles palabras y frases del usuario. Esto no se pudo lograr porque no existe un modelo acústico en español para el Sphinx2.
9. A causa de la limitación de la conclusión anterior, la aplicación es de 'comando y control', es decir, que el usuario debe decir comandos conocer comandos específicos para ejecutar una acción.
10. Se puede considerar cumplidos los objetivos de este proyecto ya que el sistema sí es funcional y representativo del modelo de negocios de restaurantes de comida rápida. En consecuencia se cumplió con los objetivos específicos de permitir la toma de pedidos y consulta de los mismos al usuario final, satisfaciendo las necesidades principales del sistema.

Recomendaciones

1. La recomendación principal para que sea factible ofrecer este sistema en un restaurante de comida rápida es explorar las limitaciones del Sphinx2 y el Festival en un ambiente de pruebas para hacer estimados para un ambiente comercial.
2. También se mejoraría bastante la experiencia del usuario al experimentar con otras voces artificiales en español porque la actual suena monótona al leer mensajes medianos y largos.

GLOSARIO

AM

Es el modelo acústico que poseen los motores de reconocimiento de voz

CENTOS

Es una distribución Linux para propósitos generales basada en RPM

GNU

Sistema operativo completo tipo Unix de software libre

Linux

Sistema operativo y núcleo. Es uno de los paradigmas del desarrollo de software libre (y de código abierto), donde el código fuente está disponible públicamente y cualquiera puede usarlo, modificarlo y/o redistribuirlo libremente.

LM

Es el modelo de lenguaje que se utiliza en los motores de reconocimiento de voz.

MYSQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario

TTS

Es una abreviatura para referirse a la conversión de texto a voz.

ANEXOS

Anexo 1: Especificaciones del Servidor

1.1 Especificaciones del CPU

```
[root@localhost proc]# cd /proc
[root@localhost proc]# cat cpuinfo
processor          : 0
vendor_id        : GenuineIntel
cpu family       : 6
model            : 15
model name       : Intel(R) Core(TM)2 Duo CPU          T5550  @ 1.83GHz
stepping         : 13
cpu MHz          : 1000.000
cache size       : 2048 KB
physical id      : 0
siblings         : 2
core id          : 0
cpu cores        : 2
apicid           : 0
fdiv_bug         : no
hlt_bug          : no
f00f_bug         : no
coma_bug         : no
fpu              : yes
fpu_exception    : yes
cpuid level      : 10
wp               : yes
flags             : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe nx lm
constant_tsc    : yes
pni              : yes
monitor          : yes
ds_cpl           : yes
est              : yes
tm2              : yes
ssse3            : yes
cx16             : yes
xtptr            : yes
lahf_lm          : yes
bogomips         : 3667.28

processor          : 1
vendor_id        : GenuineIntel
cpu family       : 6
model            : 15
model name       : Intel(R) Core(TM)2 Duo CPU          T5550  @ 1.83GHz
stepping         : 13
cpu MHz          : 1000.000
cache size       : 2048 KB
physical id      : 0
siblings         : 2
core id          : 1
cpu cores        : 2
apicid           : 1
fdiv_bug         : no
```

```
hlt_bug          : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 10
wp             : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe nx lm
constant_tsc   : 3666.98
pni            : 3666.98
monitor       : 3666.98
ds_cpl        : 3666.98
est           : 3666.98
tm2           : 3666.98
ssse3         : 3666.98
cx16          : 3666.98
xtpr          : 3666.98
lahf_lm       : 3666.98
```

1.2 Especificaciones de memoria

```
[root@localhost proc]# cd /proc
[root@localhost proc]# cat meminfo
MemTotal:      3106104 kB
MemFree:       846580 kB
Buffers:       135448 kB
Cached:        1214592 kB
SwapCached:    0 kB
Active:        1591344 kB
Inactive:      534024 kB
HighTotal:    2218816 kB
HighFree:     220680 kB
LowTotal:     887288 kB
LowFree:      625900 kB
SwapTotal:    2096440 kB
SwapFree:     2096440 kB
Dirty:        216 kB
Writeback:    0 kB
AnonPages:    775292 kB
Mapped:       98820 kB
Slab:         51628 kB
PageTables:   7260 kB
NFS_Unstable: 0 kB
Bounce:       0 kB
CommitLimit: 3649492 kB
Committed_AS: 3224704 kB
VmallocTotal: 114680 kB
VmallocUsed:  9636 kB
VmallocChunk: 104828 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
Hugepagesize: 4096 kB
```

```
[root@localhost proc]# free -t
              total          used          free          shared          buffers
cached
Mem:          3106104      2239456      866648              0      135468
1214328
-/+ buffers/cache:      889660      2216444
Swap:         2096440              0      2096440
Total:        5202544      2239456      2963088
```

```
[root@localhost proc]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda5       19G   11G   7.3G  60% /
/dev/sda7       25G   174M   24G   1% /home
tmpfs           1.5G     0   1.5G   0% /dev/shm
```

```
[root@localhost proc]# more /var/log/dmesg
Linux version 2.6.18-238.19.1.el5 (mockbuild@builder10.centos.org) (gcc
version 4.1.2 20080704 (Red Hat 4.1.2-50)) #1 SMP Fri Jul 15 07:32:29
EDT 2011
```

```
BIOS-provided physical RAM map:
```

```
BIOS-e820: 0000000000010000 - 000000000009f800 (usable)
BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
BIOS-e820: 00000000000d2000 - 00000000000d4000 (reserved)
BIOS-e820: 00000000000e0000 - 0000000000100000 (reserved)
BIOS-e820: 0000000000100000 - 00000000bf6d0000 (usable)
BIOS-e820: 00000000bf6d0000 - 00000000bf6e1000 (ACPI NVS)
BIOS-e820: 00000000bf6e1000 - 00000000c0000000 (reserved)
BIOS-e820: 00000000e0000000 - 00000000f0000000 (reserved)
BIOS-e820: 00000000fec00000 - 00000000fec10000 (reserved)
BIOS-e820: 00000000fed00000 - 00000000fed00400 (reserved)
BIOS-e820: 00000000fed14000 - 00000000fed1a000 (reserved)
BIOS-e820: 00000000fed1c000 - 00000000fed90000 (reserved)
BIOS-e820: 00000000fee00000 - 00000000fee01000 (reserved)
BIOS-e820: 00000000ff000000 - 0000000100000000 (reserved)
```

```
2166MB HIGHMEM available.
```

```
896MB LOWMEM available.
```

Anexo 2: Elementos de reconocimiento de voz

2.1 sr_server.dat

```
#!/bin/sh
exec perl -w -x $0
#!perl
# sphinx-netserver.pl
# Copyright (c) 2005 Josh McAllister
#
# This program is free software; you can redistribute it and/or modify
# it under the same terms as Perl itself.
#
# Written by Josh McAllister

use IO::Socket;
use Symbol;
use POSIX;
$SPHINXDIR='/usr/local/sphinx/share/sphinx2';
#$SPHINXDIR='/usr/local/sphinx/lib';
use Speech::Recognizer::SPX qw(:uttproc :fbs $SPHINXDIR);
#use Speech::Recognizer::SPX::Server;

# establish SERVER socket, bind and listen.
$server = IO::Socket::INET->new(LocalPort => 1069,
                                Type      => SOCK_STREAM,
                                Proto     => 'tcp',
                                Reuse     => 1,
                                Listen    => 10 )

    or die "making socket: $@\n";

# global variables
$PREFORK      = 5;          # number of children to maintain
$MAX_CLIENTS_PER_CHILD = 50; # number of clients each child
                        should process
%children     = ();        # keys are current child process
IDs
$children     = 0;         # current number of children

sub REAPER {                # takes care of dead children
    $SIG{CHLD} = \&REAPER;
    my $pid = wait;
    $children --;
    delete $children{$pid};
}
```

```

sub HUNTSMAN {
    local($SIG{CHLD}) = 'IGNORE';
    kill 'INT' => keys %children;
    exit;
}

# Fork off our children.
for (1 .. $PREFORK) {
    make_new_child();
}

# Install signal handlers.
$SIG{CHLD} = \&REAPER;
$SIG{INT} = \&HUNTSMAN;

# And maintain the population.
while (1) {
    sleep;
    for ($i = $children; $i < $PREFORK; $i++) {
        make_new_child();
    }
}

sub make_new_child {
    my $pid;
    my $sigset;

    # block signal for fork
    $sigset = POSIX::SigSet->new(SIGINT);
    sigprocmask(SIG_BLOCK, $sigset)
        or die "Can't block SIGINT for fork: $!\n";

    die "fork: $!" unless defined ($pid = fork);

    if ($pid) {
        # Parent records the child's birth and returns.
        sigprocmask(SIG_UNBLOCK, $sigset)
            or die "Can't unblock SIGINT for fork: $!\n";
        $children{$pid} = 1;
        $children++;
        return;
    } else {
        #Child
        # Child can *not* return from this subroutine.
        $SIG{INT} = 'DEFAULT';
    }
}

before

# unblock signals

```

```

sigprocmask(SIG_UNBLOCK, $sigset)
    or die "Can't unblock SIGINT for fork: $!\n";

# Initialize sphinx
fbs_init({-live          => 'FALSE',
-samp                   => 8000,
-adcin                   => 'TRUE',
-ctloffset              => 0,
-ctlcount                => 100000000,
-cepdir                  => "$SPHINXDIR/model/lm/confirm",
-datadir                 => "$SPHINXDIR/model/lm/confirm",
-agcmax                  => 'FALSE',
-langwt                  => 6.5,
-fwdfatlw                => 8.5,
-rescorelw               => 9.5,
-ugwt                    => 0.5,
-fillpen                 => 1e-10,
-silpen                  => 1e-10, #0.005,
-inspen                  => 0.65,
-top                     => 1,
-topsenfrm               => 3,
-topsenthresh           => -70000,
-beam                    => 2e-06,
-npbeam                  => 2e-06,
-lpbeam                  => 2e-05,
-lponlybeam              => 0.0005,
-nwbeam                  => 0.0005,
-fwdfat                  => 'FALSE',
-fwdfatbeam              => 1e-08,
-fwdfatnwbeam            => 0.0003,
-bestpath                => 'TRUE',
-kbdumpdir               => "$SPHINXDIR/model/lm/confirm",
-lmfn                    => "$SPHINXDIR/model/lm/confirm/confirm.lm",
-dictfn                  => "$SPHINXDIR/model/lm/confirm/confirm.dic",
-phnfn                   =>
"$SPHINXDIR/model/hmm/communicator/sphinx_2_format/phone",
-mapfn                   =>
"$SPHINXDIR/model/hmm/communicator/sphinx_2_format/map",
-hmmdir                  =>
"$SPHINXDIR/model/hmm/communicator/sphinx_2_format",
-hmmdirlist              =>
"$SPHINXDIR/model/hmm/communicator/sphinx_2_format",
-ndictfn                 =>
"$SPHINXDIR/model/hmm/communicator/sphinx_2_format/noisedict",
'-8bsen'                 => 'TRUE',
-sendumpfn               =>
"$SPHINXDIR/model/hmm/communicator/sphinx_2_format/sendump",
-cbdir                   => "$SPHINXDIR/model/hmm/communicator"});

```

```

# handle connections until we've reached $MAX_CLIENTS_PER_CHILD
for ($i=0; $i < $MAX_CLIENTS_PER_CHILD; $i++) {
    my $buf = undef;
    $client = $server->accept() or last;
    uttproc_begin_utt();
    my $count = 0;
    my $datasize = readline $client;
    chomp $datasize;
    my $b = read ($client, my($buf), $datasize);
    #print "SERVER DEBUG: Expecting $datasize bytes, got $b
bytes.\n";
    uttproc_rawdata($buf, 1) or die "uttproc_rawdata failed";
    uttproc_end_utt();
    my ($fr, $hyp) = uttproc_result(1);
    #print "frames $fr\n";
    print STDERR "SERVER RESULT: $hyp\n";
    print $client "$hyp";
    close $client;
}
# tidy up gracefully and finish

# this exit is VERY important, otherwise the child will become
# a producer of more and more children, forking yourself into
# process death.
fbs_end();
exit;
}
}

```

2.2 sr_client.dat

```

#!/usr/bin/perl
# sphinx-netclient.pl
# Copyright (c) 2005 Josh McAllister
#
# This program is free software; you can redistribute it and/or modify
# it under the same terms as Perl itself.
#
# Written by Josh McAllister

#die ("usage: $0 \n") if not -e $ARGV[0];

print "Result: " . asr("$ARGV[0]") . "\n";
exit;

sub asr {

```



```

use IO::Socket;
use FileHandle;
use IPC::Open2;
my $file = shift or return undef;
my $host = shift || 'localhost';
my $port = shift || '1069';
my $fh;

my $remote = IO::Socket::INET->new(
    Proto    => "tcp",
    PeerAddr => "$host",
    PeerPort => "$port",
    ) or return undef;

#Idea here being that you can pass a reference to an existing file
handle... not yet implemented, just pass a filename.
if (ref $file) {
    my $fh = $file;
} else {
    open (FH, $file) || return undef;
    $fh = *FH;
}

$file =~ /(gsm|wav)$/;
my $type = $1;
if ($type !~ /gsm|wav/) {
    warn "Unknown file type ($file)";
    return undef;
}
#print "FTYPE: $type\n";
$pid = open2(*SOXIN, *SOXOUT, "sox -t $type - -s -r 8000 -w -t wav -
2>/dev/null") || warn ("Could not open2.\n");

binmode $fh;
binmode SOXIN;
binmode SOXOUT;
binmode $remote;

while (defined(my $b = read $fh, my($buf), 4096)) {
    last if $b == 0;
    $count += $b;
    print SOXOUT $buf;
}
close SOXOUT;

$count = 0;
my $sox = undef;
while (defined(my $b = read SOXIN, my($buf), 4096)) {
    last if $b == 0;

```

```
    $count += $b;
    $sox .= $buf;
}
print $remote length($sox) . "\n";
print $remote "$sox";
close SOXIN;
#print "DEBUG: Waiting for result.\n";
$count=0;
while (defined(my $b = read $remote, my($buf), 4096)) {
    last if $b == 0;
    $count += $b;
    $result .= $buf;
}
close $fh;
close $remote;
return "$result";
}
```

2.3 confirm.vocab

CANCEHLAR
CANCEL
CANCELAR
CHAO
MEHNOO
MEHNUU
MENUU
NO
NOH
NOO
NOP
OHK
OHKAY
OK
OKAY
SAHLEER
SAHLIR
SEE
SI
SIH
ZEE

Anexo 3: Código de la base de datos

-- TABLE: Categoria

```
CREATE TABLE Categoria(  
  Categoria_id INT AUTO_INCREMENT,  
  Nombre VARCHAR(50),  
  Descripcion VARCHAR(100),  
  Activo SMALLINT,  
  PRIMARY KEY (Categoria_id)  
)ENGINE=MYISAM;
```

-- TABLE: cdr

```
CREATE TABLE cdr(  
  Llamada_id INT AUTO_INCREMENT,  
  calldate DATETIME,  
  clid CHAR(80),  
  src VARCHAR(80),  
  dst VARCHAR(80),  
  dcontext VARCHAR(80),  
  channel VARCHAR(80),  
  dstchannel VARCHAR(80),  
  lastapp VARCHAR(80),  
  lastdata VARCHAR(80),  
  duration INT,  
  billsec INT,  
  disposition VARCHAR(45),  
  amaflags INT,  
  accountcode VARCHAR(20),  
  userfield VARCHAR(255),  
  uniqueid VARCHAR(32),  
  PRIMARY KEY (Llamada_id)  
)ENGINE=MYISAM;
```

-- TABLE: Ciudad

```
CREATE TABLE Ciudad(  
  Ciudad_id INT AUTO_INCREMENT,  
  Nombre VARCHAR(50),  
  Activo SMALLINT,  
  PRIMARY KEY (Ciudad_id)  
)ENGINE=MYISAM;
```

-- TABLE: Ciudadela

```
CREATE TABLE Ciudadela(  
  Ciudadela_id INT AUTO_INCREMENT,  
  Nombre VARCHAR(50),  
  Indicaciones VARCHAR(100),  
  Activo SMALLINT,  
  Ciudad_id INT NOT NULL,  
  PRIMARY KEY (Ciudadela_id)  
)ENGINE=MYISAM;
```

-- TABLE: Combo

```
CREATE TABLE Combo(  
  Combo_id INT AUTO_INCREMENT,  
  Nombre VARCHAR(50),  
  Descripcion VARCHAR(100),  
  Activo SMALLINT,  
  PRIMARY KEY (Combo_id)  
)ENGINE=MYISAM;
```

-- TABLE: Combo_item

```
CREATE TABLE Combo_item(  
  Combo_Item_id INT AUTO_INCREMENT,  
  Activo SMALLINT,  
  Combo_id INT NOT NULL,  
  Item_id INT NOT NULL,  
  PRIMARY KEY (Combo_Item_id)  
)ENGINE=MYISAM;
```

-- TABLE: Direccion

```
CREATE TABLE Direccion(  
  Direccion_id INT AUTO_INCREMENT,  
  Nombre VARCHAR(50),  
  Indicaciones VARCHAR(100),  
  Activo SMALLINT,  
  Ciudadela_id INT NOT NULL,  
  PRIMARY KEY (Direccion_id)  
)ENGINE=MYISAM;
```

-- TABLE: Estado_Pedido

```
CREATE TABLE Estado_Pedido(  
  Estado_Pedido_id INT AUTO_INCREMENT,  
  Nombre VARCHAR(50),  
  Descripcion VARCHAR(100),  
  Activo SMALLINT,  
  PRIMARY KEY (Estado_Pedido_id)  
)ENGINE=MYISAM;
```

-- TABLE: Item

```
CREATE TABLE Item(  
  Item_id INT AUTO_INCREMENT,  
  Nombre VARCHAR(50),  
  Descripcion VARCHAR(100),  
  Activo SMALLINT,  
  Precio DECIMAL(10, 2),  
  Categoria_id INT NOT NULL,  
  PRIMARY KEY (Item_id)  
)ENGINE=MYISAM;
```

-- TABLE: Llamada_calificacion

```
CREATE TABLE Llamada_calificacion(  
  Llamada_calificacion_id INT NOT NULL,
```

```
Llamada_id      VARCHAR(32),
Calificacion    INT,
Persona_id      INT,
PRIMARY KEY (Llamada_calificacion_id)
)ENGINE=MYISAM;
```

-- TABLE: Pedido

```
CREATE TABLE Pedido(
  Pedido_id      INT          AUTO_INCREMENT,
  Nombre         VARCHAR(50),
  Descripcion    VARCHAR(100),
  Activo         SMALLINT,
  Fecha_Creacion DATETIME    NOT NULL,
  Precio_total   DECIMAL(10, 2) NOT NULL,
  Precio_iva     DECIMAL(10, 2),
  Precio_final   DECIMAL(10, 2),
  Estado_Pedido_id INT        NOT NULL,
  Direccion_id  INT          NOT NULL,
  Persona_id     INT          NOT NULL,
  PRIMARY KEY (Pedido_id)
)ENGINE=MYISAM;
```

-- TABLE: Pedido_Detalle

```
CREATE TABLE Pedido_Detalle(
  Pedido_Detalle_id INT        AUTO_INCREMENT,
  Nombre            VARCHAR(50),
  Activo           SMALLINT,
  Unidades         INT,
  Precio_unidad    DECIMAL(10, 2) NOT NULL,
  Precio_total     DECIMAL(10, 2),
  Precio_iva       DECIMAL(10, 2),
  Precio_final     DECIMAL(10, 2),
  Item_id          INT          NOT NULL,
  Combo_id         INT          NOT NULL,
  Pedido_id        INT          NOT NULL,
  PRIMARY KEY (Pedido_Detalle_id)
)ENGINE=MYISAM;
```

-- TABLE: Persona

```
CREATE TABLE Persona(  
  Persona_id      INT      AUTO_INCREMENT,  
  Cedula          VARCHAR(11),  
  Nombre1         VARCHAR(50) NOT NULL,  
  Nombre2         VARCHAR(100),  
  Apellido1       VARCHAR(100) NOT NULL,  
  Apellido2       VARCHAR(50),  
  Fecha_Registro  DATETIME,  
  Fecha_Nacimiento DATETIME,  
  Password        VARCHAR(10),  
  Persona_Tipo_id INT,  
  PRIMARY KEY (Persona_id)  
)ENGINE=MYISAM;
```

-- TABLE: Persona_Direccion

```
CREATE TABLE Persona_Direccion(  
  Persona_Direccion_id INT      AUTO_INCREMENT,  
  Activo                SMALLINT,  
  Persona_id            INT      NOT NULL,  
  Direccion_id          INT      NOT NULL,  
  PRIMARY KEY (Persona_Direccion_id)  
)ENGINE=MYISAM;
```

-- TABLE: Persona_Tipo

```
CREATE TABLE Persona_Tipo(  
  Persona_Tipo_id INT      AUTO_INCREMENT,  
  Activo          SMALLINT NOT NULL,  
  Nombre          VARCHAR(50),  
  PRIMARY KEY (Persona_Tipo_id)  
)ENGINE=MYISAM;
```

```
ALTER TABLE Ciudadela ADD CONSTRAINT RefCiudad5
  FOREIGN KEY (Ciudad_id)
  REFERENCES Ciudad(Ciudad_id);
```

```
ALTER TABLE Combo_item ADD CONSTRAINT RefCombo3
  FOREIGN KEY (Combo_id)
  REFERENCES Combo(Combo_id);
```

```
ALTER TABLE Combo_item ADD CONSTRAINT RefItem4
  FOREIGN KEY (Item_id)
  REFERENCES Item(Item_id);
```

```
ALTER TABLE Direccion ADD CONSTRAINT RefCiudadela6
  FOREIGN KEY (Ciudadela_id)
  REFERENCES Ciudadela(Ciudadela_id);
```

```
ALTER TABLE Item ADD CONSTRAINT RefCategoria1
  FOREIGN KEY (Categoria_id)
  REFERENCES Categoria(Categoria_id);
```

```
ALTER TABLE Llamada_calificacion ADD CONSTRAINT RefPersona33
  FOREIGN KEY (Persona_id)
  REFERENCES Persona(Persona_id);
```

```
ALTER TABLE Pedido ADD CONSTRAINT RefDireccion28
  FOREIGN KEY (Direccion_id)
  REFERENCES Direccion(Direccion_id);
```

```
ALTER TABLE Pedido ADD CONSTRAINT RefPersona29
  FOREIGN KEY (Persona_id)
  REFERENCES Persona(Persona_id);
```

```
ALTER TABLE Pedido ADD CONSTRAINT RefEstado_Pedido14
  FOREIGN KEY (Estado_Pedido_id)
  REFERENCES Estado_Pedido(Estado_Pedido_id);
```



```
ALTER TABLE Pedido_Detalle ADD CONSTRAINT RefItem11
  FOREIGN KEY (Item_id)
  REFERENCES Item(Item_id);
```

```
ALTER TABLE Pedido_Detalle ADD CONSTRAINT RefCombo12
  FOREIGN KEY (Combo_id)
  REFERENCES Combo(Combo_id);
```

```
ALTER TABLE Pedido_Detalle ADD CONSTRAINT RefPedido13
  FOREIGN KEY (Pedido_id)
  REFERENCES Pedido(Pedido_id);
```

```
ALTER TABLE Persona ADD CONSTRAINT RefPersona_Tipo31
  FOREIGN KEY (Persona_Tipo_id)
  REFERENCES Persona_Tipo(Persona_Tipo_id);
```

```
ALTER TABLE Persona_Direccion ADD CONSTRAINT RefPersona26
  FOREIGN KEY (Persona_id)
  REFERENCES Persona(Persona_id);
```

```
ALTER TABLE Persona_Direccion ADD CONSTRAINT RefDireccion27
  FOREIGN KEY (Direccion_id)
  REFERENCES Direccion(Direccion_id);
```

Anexo 4: extensions.conf

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

[asr]
;Bienvenida
exten => 123, 1, Answer()
exten => 123, 2, AGI(bienvenida.agi)
exten => 123, 3, Hangup()

;Menu principal
exten => 101, 1, AGI(menu_principal.agi)

;Mas informacion
exten => 102, 1 AGI(mas_informacion.agi)

;Revisar estado de pedidos
exten => 103, 1, AGI(revisar_estado_pedidos.agi)

;Menu de Categorias
exten => 104, 1, AGI(select_categoria.agi)

;Extensión para comunicarse con un operador humano
exten => 789, 1, Answer()
exten => 789, 2, Verbose('En la extension 789 para comunicarse con un
operador humano')
exten => 789, 3, Hangup()
```

Anexo 5: Scripts AGI

Bienvenida.agi

```
#!/bin/sh
exec perl -w -x $0
#!/perl
$|=1; #borrar buffer

use Asterisk::AGI;
my $AGI = new Asterisk::AGI;

%input = $AGI->ReadParse();

#Utilizar text2wave de Festival para contestarle al cliente.
sub decir {
    use File::Basename;
    use Digest::MD5 qw(md5_hex);

    my($text) = @_ ;
    my $hash = md5_hex($text);
    my $sounddir = "/var/lib/asterisk/sounds/tts";
    my $wavefile = "$sounddir/". "tts-$hash.wav";
    my $t2wp= "/usr/bin/";

    unless (-f $wavefile) {
        open(fileOUT, ">$sounddir."/say-text-$hash.txt");
        print fileOUT "$text";
        close(fileOUT);
        my $execf=$t2wp."text2wave -F 8000 -o $wavefile
$sounddir/say-text-$hash.txt > /dev/null";
        system($execf);
        unlink($sounddir."/say-text-$hash.txt");
    }
    $AGI->stream_file('tts/'.basename($wavefile, ".wav"));
    unlink($wavefile);
}#end sub decir

decir("Hola! Bienvenido a Macvoip. ");
my $cedula=$AGI->get_variable('cedula'); #Reviso si esta
loggeado
$AGI->verbose("En bienvenida, cedula: $cedula");

if($cedula==""){ #Si no esta loggeado..
    $AGI->verbose("Va a autenticar cliente.");
    $AGI->exec("AGI autenticar_cliente.agi");
```

```

        return undef;
    }else{
        $AGI->verbose("Va a autenticar cliente.");
        $AGI->exec("AGI menu_principal.agi");
        return undef;
    }
}

```

Autenticar_cliente.agi

```

#!/bin/sh
exec perl -w -x $0
#!/perl
$|=1; #borrar buffer

use Asterisk::AGI;
my $AGI = new Asterisk::AGI;

%input = $AGI->ReadParse();

# Copyright (c) 2005 Josh McAllister
# This program is free software; you can redistribute it and/or modify
it under the same terms as Perl itself.
sub asr {
    use IO::Socket;
    use FileHandle;
    use IPC::Open2;

    my $file = shift or return undef;
    my $host = shift || 'localhost';
    my $port = shift || '1069'; # puerto del server
    my $fh;

    my $remote = IO::Socket::INET->new(
        Proto => "tcp",
        PeerAddr => "$host",
        PeerPort => "$port",
    ) or return undef;

    if (ref $file) {
        my $fh = $file;
    } else {
        open (FH, $file) || return undef;
        $fh = *FH;
    }

    $file =~ /(gsm|wav)$/;
    my $type = $1;

```

```

    if ($type !~ /gsm|wav/) {
        warn "Unknown file type ($file)";
        return undef;
    }

    #print "FTYPE: $type\n";
    $pid = open2(*SOXIN, *SOXOUT, "sox -t $type - -s -r 8000 -w -t
wav - 2>/dev/null") || warn ("Could not open2.\n");

    binmode $fh;
    binmode SOXIN;
    binmode SOXOUT;
    binmode $remote;

    while (defined(my $b = read $fh, my($buf), 4096)) {
        last if $b == 0;
        $count += $b;
        print SOXOUT $buf;
    }
    close SOXOUT;

    $count = 0;
    my $sox = undef;
    while (defined(my $b = read SOXIN, my($buf), 4096)) {
        last if $b == 0;
        $count += $b;
        $sox .= $buf;
    }
    print $remote length($sox) . "\n";
    print $remote "$sox";
    close SOXIN;

    #print "DEBUG: Waiting for result.\n";

    $count=0;
    while (defined(my $b = read $remote, my($buf), 4096)) {
        last if $b == 0;
        $count += $b;
        $result .= $buf;
    }

    close $fh;
    close $remote;

    return "$result";
}#end sub asr

# Crea una conexi3n a la base de datos.
sub conectarBase {

```

```

        use strict;
        use DBI;
        my $dbh = DBI->connect(          # DataBase Handle que maneja la
conexión a MySQL
            'DBI:mysql:asterisk:localhost:3306',
            'root',
            'root',
            {RaiseError => 1}
        ) or die "No se pudo conectar a MySQL: $DBI::errstr\n";

        return $dbh;
    }#end sub conectarBase

#Utilizar text2wave de Festival para contestarle al cliente.
sub decir {
    use File::Basename;
    use Digest::MD5 qw(md5_hex);

    my($text) = @_ ;
    my $hash = md5_hex($text);
    my $sounddir = "/var/lib/asterisk/sounds/tts";
    my $wavefile = "$sounddir/". "tts-$hash.wav";
    my $t2wp= "/usr/bin/";

    unless (-f $wavefile) {
        open(fileOUT, ">$sounddir"."/say-text-$hash.txt");
        print fileOUT "$text";
        close(fileOUT);
        my $execf=$t2wp."text2wave -F 8000 -o $wavefile
$sounddir/say-text-$hash.txt > /dev/null";
        system($execf);
        unlink($sounddir."/say-text-$hash.txt");
    }
    $AGI->stream_file('tts/'.basename($wavefile, ".wav"));
    unlink($wavefile);
}#end sub decir

# Revisa si la cedula recibida existe en la base de datos.
sub getExisteCedula{
    my($dbh, $cedula) = @_ ;

    my $query="select count(*) from Persona where Cedula =
'$cedula'";
    my $sth = $dbh->prepare($query);
    $sth->execute;
    my $cedula_registrada=$sth->fetchrow_array();

    return $cedula_registrada;
}#end sub getExisteCedula

```

```

# Agrega una Persona en la base de datos
sub registrarCedulaPersona{
    my ($dbh, $cedula, $password) = @_;
    use Mysql;

    my $host = "localhost";
    my $database = "asterisk";
    my $user = "root";
    my $pw = "root";
    my $id_tipo_cliente=1;

    my $connect = Mysql->connect($host, $database, $user, $pw);
    $connect->selectdb($database);

    my $query="insert into Persona(Fecha_Registro, Password,
Persona_Tipo_id, Cedula)";
    $query.=" values(NOW(), '$password', $id_tipo_cliente,
'$cedula');"
    $AGI->verbose("En registrarCedulaPersona, query: $query");
    my $execute = $connect->query($query);
    my $lastid = $execute->insertid($query);

    return $lastid;
}#end sub registrarCedulaPersona

# Pregunta al usuario si desea registrarse. Si este desea registrarse,
se solicita al cliente un numero de cÃ©dula y contraseÃ±a.
sub askDeseaRegistrarse {
    my($max_tries, $max_time_record, $beep_file, $wait_segundos,
$min_digitos_pass, $max_digitos_pass, $max_tries_pass) = @_;

    decir("Desea registrarse como usuario?");
    my $tries=0;
    my $vresponse = '';
    while ($tries < $max_tries) {
        $vresponse = '';

        $AGI->stream_file($beep_file, '');
        $AGI->record_file("/tmp/$$", 'wav', '0', $max_time_record);
        #0? 3s
        $vresponse = asr("/tmp/$$.wav");
        $AGI->verbose("# Intento responderDeseaRegistrarse: $tries,
vresponse: $vresponse");

        if ($vresponse =~ /CHAO/i){
            $AGI->exec("AGI chao.agi");
            return undef;
        }
    }
}

```

```

        }elseif ($vresponse =~
/CANCELAR|CANCEL|CANCEHLAR|NO|NOH|NOP/i) {
            $AGI->verbose("Va al Menu principal");
            $AGI->exec("AGI menu_principal.agi");
            return undef;
        }elseif ($vresponse =~ /SEE|SIH|SI|OK|OHK|OKAY|OHKAY/i){
            my $tries=0;
            my $pass1="";
            while($tries<$max_tries_pass){
                #Repetir
                #Se acabaron los intentos de setear pass. Ir a Menu
                #para q no se vaya
                #ambas si se equivoca
                decir("Por favor ingrese una contrasenia de
entre $min_digitos_pass y $max_digitos_pass diigitos. ");
                $pass1 = $AGI->get_data($beep_file,
$wait_segundos, $max_digitos_pass);
                #
                $AGI->verbose("pass1: $pass1");
                if(length($pass1)>=$min_digitos_pass &&
length($pass1)<=$max_digitos_pass){
                    decir("Por favor confirme la contrasenia
ingresada. ");
                    my $pass2 = $AGI->get_data($beep_file,
$wait_segundos, $max_digitos_pass);

                    if($pass1==$pass2){
                        $AGI->verbose("Contrasenia
confirmada correctamente");
                        decir("Contrasenia confirmada
correctamente. ");
                        my
                        $persona_ingresada=registrarCedulaPersona($dbh, $cedula, $pass1);
                        if($persona_ingresada){
                            decir("Usuario registrado
exitosamente.");
                            $AGI->verbose("Usuario
registrado exitosamente");
                            $AGI->set_variable('cedula',
$cedula);
                            $AGI->verbose("Va al Menu
principal");
                            $AGI->exec("AGI
menu_principal.agi");
                            return undef;
                        }
                    }
                    $tries++;
                }
            }
        }
    }
}
#
$tries++;

```



```

        } #end while
    } #end sub askDeseaRegistrarse

sub autenticarCedula{
    my($dbh, $cedula, $password) = @_ ;

    my $query="select count(*) from Persona where Cedula = '$cedula'
and Password='$password'";
    my $sth = $dbh->prepare($query);
    $sth->execute;
    my $cedula_autenticada=$sth->fetchrow_array();

    return $cedula_autenticada;
} #end sub autenticarCedula

# Recorre la cedula para decir cada numero
sub decirCedula{
    my($cedula) = @_ ;
    $AGI->verbose("En decirCedula");

    @nums = split(//, $cedula);
    foreach $numero (@nums) {
        decir($numero);
        $AGI->verbose("numero: $numero");
    }
}

} # end sub decirCedula

# Extrae de la base el nombre de la persona con el numero de cedula
especificado.
sub getNombre{
    my($dbh, $cedula) = @_ ;
    my $query="SELECT Nombre1 FROM Persona where Cedula='$cedula'";
    my $sth = $dbh->prepare($query);
    $sth->execute;
    my $arr = $sth->fetchrow_array;

    return $arr;
} # end sub getNombre

my $dbh=conectarBase();
if($dbh){
    my $max_tries=2;
    my $max_time_record=3000;
    my $beep_file='beep';
    my $wait_segundos=3500;

```

```

my $min_digitos_pass=4;
my $max_digitos_pass=8;
my $max_tries_pass=2;
my $num_digitos_cedula=10;

decir("Ingrese su numero de cedula para autenticarse.");

my $tries=0;
while ($tries < $max_tries) {
    $cedula = $AGI->get_data($beep_file, $wait_segundos,
$num_digitos_cedula);

    if(length($cedula)==$num_digitos_cedula){
        decir("Ingreso la cedula: ");
        decirCedula($cedula);
        my $cedula_registrada=getExisteCedula($dbh, $cedula);
        $AGI->verbose("cedula: $cedula, cedula_registrada:
$cedula_registrada");

        if($cedula_registrada){           #ingresar contraseña
            $AGI->verbose("Cedula registrada. Ingresar
contrasenia");

            decir("Ingrese su contraseña.");
            my $tries=0;
            my $password="";
            while($tries<$max_tries_pass){
                $password = $AGI->get_data($beep_file,
$wait_segundos, $max_digitos_pass);
                if(length($password)>=$min_digitos_pass
&& length($password)<=$max_digitos_pass){
                    my
$cedula_autenticada=autenticarCedula($dbh, $cedula, $password);
                    $AGI->verbose("cedula_autenticada:
$cedula_autenticada");

                    if($cedula_autenticada>0){
                        my
$primer_nombre=getNombre($dbh, $cedula);
                        if($primer_nombre){
                            decir("Bienvenido,
$primer_nombre.");
                        }else{
                            decir("Bienvenido!");
                        }
                        $AGI->set_variable('cedula',
$cedula);
                        #
                        $AGI->verbose("Va al Menu
principal");
                        $AGI->exec("AGI
menu_principal.agi");
                    }
                }
            }
        }
    }
}

```

```

                                return undef;
                                }else{
                                decir("Usuario no
autenticado.");
                                }
                                }
                                $tries++;
                                } #seguir
                                }else{ #registrar
usuario
                                askDeseaRegistrarse($max_tries,
$max_time_record, $beep_file, $wait_segundos, $min_digitos_pass,
$max_digitos_pass, $max_tries_pass, $cedula);
                                return undef;
                                }
                                $tries = $max_tries;
                                }elsif(length($cedula)>0){
                                decir("Su numero de cedula estaa incompleto. ");
                                $tries=0;
                                }
                                $tries++;
                                } #end while intentos
# decir("Iraa al menuu principal");
$AGI->verbose("Va al Menu principal");
$AGI->exec("AGI menu_principal.agi");
return undef;
}else{ #Error conexion
decir("Ocurrioo un error. SerÃ; redireccionado a un operador.");
$AGI->verbose("Va a hablar con un operador.");

$AGI->set_context('asr');
$AGI->set_extension('789');
$AGI->set_priority(1);
return undef;
}
return undef;

```

menu_principal.agi

```

#!/bin/sh
exec perl -w -x $0
#!/perl
$|=1; #borrar buffer

use Asterisk::AGI;
my $AGI = new Asterisk::AGI;

```

```



```

```

close SOXOUT;

$count = 0;
my $sox = undef;
while (defined(my $b = read SOXIN, my($buf), 4096)) {
    last if $b == 0;
    $count += $b;
    $sox .= $buf;
}
print $remote length($sox) . "\n";
print $remote "$sox";
close SOXIN;

#print "DEBUG: Waiting for result.\n";

$count=0;
while (defined(my $b = read $remote, my($buf), 4096)) {
    last if $b == 0;
    $count += $b;
    $result .= $buf;
}

close $fh;
close $remote;

return "$result";
}#end sub asr

sub conectarBase {
    use strict;
    use DBI;
    my $dbh = DBI->connect(          # DataBase Handle que maneja la
conexiÃ³n a MySQL
        'DBI:mysql:asterisk:localhost:3306',
        'root',
        'root',
        {RaiseError => 1}
    ) or die "No se pudo conectar a MySQL: $DBI::errstr\n";

    return $dbh;
}#end sub conectarBase

sub decir {
    use File::Basename;
    use Digest::MD5 qw(md5_hex);

    my($text) = @_ ;
    my $hash = md5_hex($text);
    my $sounddir = "/var/lib/asterisk/sounds/tts";

```

```

my $wavefile = "$sounddir/" . "tts-$hash.wav";
my $t2wp= "/usr/bin/";

unless (-f $wavefile) {
    open(fileOUT, ">$sounddir"/say-text-$hash.txt");
    print fileOUT "$text";
    close(fileOUT);
    my $execf=$t2wp."text2wave -F 8000 -o $wavefile
$sounddir/say-text-$hash.txt > /dev/null";
    system($execf);
    unlink($sounddir."/say-text-$hash.txt");
}
$AGI->stream_file('tts/' . basename($wavefile, ".wav"));
unlink($wavefile);
}#end sub decir

sub getNumPedidosNoTerminados{
    my($dbh, $cedula, $estado_terminado_id) = @_ ;

    my $query="select count(*)";
    $query.=" from Pedido p join Estado_Pedido ep on
p.Estado_pedido_id=ep.Estado_pedido_id join Persona per on
p.Persona_id=per.Persona_id";
    $query.=" where per.Cedula='$cedula' and
p.Estado_pedido_id!=$estado_terminado_id and p.Activo=1;";

    my $sth = $dbh->prepare($query);
    $sth->execute;
    my $resultado=$sth->fetchrow_array();

    return $resultado;
}#end sub getNumPedidosNoTerminados

sub getMenuPrincipalOpciones{
    my($cedula, $tmp_file_noconf, $dbh) = @_ ;
    my %opciones_menu_principal = ();

    $opciones_menu_principal{0} = "Menu de categoriias";
    if($cedula==""){
        $opciones_menu_principal{4} = "Autenticarse o registrarse";
    }else{
        my $length_cedula=length($cedula);
        if($length_cedula==10){
            if($tmp_file_noconf != ""){ #si hay un tmp, hay
pedido por confirmar.
                $opciones_menu_principal{1} = "Confirmar su
pedido";
            }
        }
    }
}

```

```

        if($dbh){
            my $num_pedidos_noterminados =
getNumPedidosNoTerminados($dbh, $cedula, 6);
            $AGI->verbose("num_pedidos_noterminados:
$num_pedidos_noterminados");

            if($num_pedidos_noterminados==1){
                $opciones_menu_principal{2} = "Revisar
estado del pedido pendiente";
                #
                $opciones_menu_principal{3} = "Revisar
detalles del pedido pendiente";
            }elseif($num_pedidos_noterminados>1){
                $opciones_menu_principal{2} = "Revisar
estados de los $num_pedidos_noterminados pedidos pendientes";
                #
                $opciones_menu_principal{3} = "Revisar
detalles de los $num_pedidos_noterminados pedidos pendientes";
            }
        }
    }
    $opciones_menu_principal{5} = "Mas informacion";

    return %opciones_menu_principal;
} #end sub getMenuPrincipalOpciones

my $digitos=10;
my $max_tries=2;
my $max_time_record=3000;
my $beep_file='beep';
my $wait_segundos=3500;

decir("Menuu principal. Diga okey al ooir la opcioon a seleccionar. ");

my $dbh=conectarBase();
if($dbh){
    my $tmp_file_noconf=$AGI->get_variable('tmp_file_noconf');
    $cedula=$AGI->get_variable('cedula');
    $AGI->verbose("Luego en Menu principal, cedula: $cedula ");
    my %opciones_menu_principal= getMenuPrincipalOpciones($cedula,
$tmp_file_noconf, $dbh);

    $AGI->verbose("Opciones Menu principal: ");
    while (($opcion_id, $nombre_opcion) =
each(%opciones_menu_principal)){
        $AGI->verbose("opcion_id: $opcion_id, nombre_opcion:
$nombre_opcion");
        decir($nombre_opcion);
    }
}

```

```

$tries=0;
while ($tries < $max_tries){
    $vresponse = '';
    $AGI->stream_file($beep_file,'');
    $AGI-
>record_file("/tmp/$$", 'wav', '0', $max_time_record);
    $vresponse = asr("/tmp/$$wav");

    $AGI->verbose("# Intento elegirOpcion: $tries,
vresponse: $vresponse");

    if ($vresponse =~ /CHAO|CANCELAR|CANCEL|CANCEHLAR/i)
    {
        $AGI->exec("AGI chao.agi");
        return undef;
    }elseif ($vresponse =~
/SEE|SIH|SI|OK|OHK|OKAY|OHKAY/i) {
        $AGI->verbose("opcion_id: $opcion_id");

        if($opcion_id==0) {
            #Menu de
Categorías
            $AGI->verbose("Va al Menu de
Categorías.");
            $AGI->exec("AGI select_categoria.agi");
            return undef;
        }elseif($opcion_id==1){
            $AGI->verbose("Va a Confirmar Pedido.");
            $AGI->exec("AGI confirmar_pedido.agi");
            return undef;
        }elseif($opcion_id==2){
            $AGI->verbose("Va a Revisar estado del
pedido.");
            $AGI->exec("AGI
revisar_estado_pedidos.agi");
            return undef;
        }elseif($opcion_id==3){
            $AGI->verbose("Va a Revisar detalle
pedidos pendientes.");
        }elseif($opcion_id==4){
            $AGI->verbose("Va a autenticarse o
registrarse");
            $AGI->exec("AGI autenticar_cliente.agi");
            return undef;
        }elseif($opcion_id==5){
            $AGI->verbose("Va a Mas Informacion.");
            $AGI->exec("AGI mas_informacion.agi");
            return undef;
        }
    }
    return 1;
}

```



```

        }
        $tries++;
    } #end while intentos
} #end while opciones del menu principal

#Terminã³ el listado de opciones y no ha seleccionado ninguna

decir("No seleccionoo ninguna opcioon. Se repetiraa el Menuu
principal.");
decir("Diga Cancelar, noo o chao para terminar la llamada.");

$tries=0;
while ($tries < $max_tries){
    $vresponse = '';
    $AGI->stream_file($beep_file, '');
    $AGI->record_file("/tmp/$$, 'wav', '0', $max_time_record);
    $vresponse = asr("/tmp/$$.wav");

    $AGI->verbose("# Intento noRepetirMenuPrincipal: $tries,
vresponse: $vresponse");

    if ($vresponse =~ /CANCELAR|CANCEL|CHAO|CANCEHLAR/i) {
#Ir al comienzo del menu..?
        $AGI->exec("AGI chao.agi");
        return undef;
    }elseif ($vresponse =~ /SEE|SIH|SI|OK|OHK|OKAY|OHKAY/i){
        $AGI->set_context('asr');        #Menu principal
        $AGI->set_extension('101');
        $AGI->set_priority(1);
        return 1;
    }
    $tries++;
} #end while intentos

$AGI->set_context('asr');        #Menu principal
$AGI->set_extension('101');
$AGI->set_priority(1);
return 1;
}else{        #Error conexion a la base
decir("Ocurrioo un error. Serã; redireccionado a un operador.");
$AGI->verbose("Va a hablar con un operador.");

$AGI->set_context('asr');
$AGI->set_extension('789');
$AGI->set_priority(1);
return undef;
}
}

```

Select_categoria.agi

```
#!/bin/sh
exec perl -w -x $0
#!/perl
$|=1; #borrar buffer

use Asterisk::AGI;
my $AGI = new Asterisk::AGI;

%input = $AGI->ReadParse();

# Copyright (c) 2005 Josh McAllister
# This program is free software; you can redistribute it and/or modify
it under the same terms as Perl itself.
sub asr {
    use IO::Socket;
    use FileHandle;
    use IPC::Open2;

    my $file = shift or return undef;
    my $host = shift || 'localhost';
    my $port = shift || '1069'; # puerto del server
    my $fh;

    my $remote = IO::Socket::INET->new(
        Proto => "tcp",
        PeerAddr => "$host",
        PeerPort => "$port",
    ) or return undef;

    #Idea here being that you can pass a reference to an existing
file handle... not yet implemented, just pass a filename.
    if (ref $file) {
        my $fh = $file;
    } else {
        open (FH, $file) || return undef;
        $fh = *FH;
    }

    $file =~ /(gsm|wav)$/;
    my $type = $1;
    if ($type !~ /gsm|wav/) {
        warn "Unknown file type ($file)";
        return undef;
    }

    #print "FTYPE: $type\n";
```

```

    $pid = open2(*SOXIN, *SOXOUT, "sox -t $type - -s -r 8000 -w -t
wav - 2>/dev/null") || warn ("Could not open2.\n");

    binmode $fh;
    binmode SOXIN;
    binmode SOXOUT;
    binmode $remote;

    while (defined(my $b = read $fh, my($buf), 4096)) {
        last if $b == 0;
        $count += $b;
        print SOXOUT $buf;
    }
    close SOXOUT;

    $count = 0;
    my $sox = undef;
    while (defined(my $b = read SOXIN, my($buf), 4096)) {
        last if $b == 0;
        $count += $b;
        $sox .= $buf;
    }
    print $remote length($sox) . "\n";
    print $remote "$sox";
    close SOXIN;

    #print "DEBUG: Waiting for result.\n";

    $count=0;
    while (defined(my $b = read $remote, my($buf), 4096)) {
        last if $b == 0;
        $count += $b;
        $result .= $buf;
    }

    close $fh;
    close $remote;

    return "$result";
}#end sub asr

# Crea una conexi3n a la base de datos. Retorna un DataBaseHandle que
maneja la conexi3n a MySQL
sub conectarBase{
    use strict;
    use DBI;
    my $dbh = DBI->connect(          # DataBase Handle que maneja la
conexi3n a MySQL
        'DBI:mysql:asterisk:localhost:3306',

```

```

        'root',
        'root',
        {RaiseError => 1}
    ) or die "No se pudo conectar a MySQL: $DBI::errstr\n";

    return $dbh;
}#end sub conectarBase

sub getCuantasCategorias{
    my($dbh) = @_ ;
    my $query="SELECT count(*) FROM Categoria where Activo=1 and
Descripcion!='';";
    my $sth = $dbh->prepare($query);
    $sth->execute;
    my $arr = $sth->fetchrow_array;

    return $arr;
} #end sub getCuantasCategorias

# Retorna el total de categorías activas.
sub getCategorias {
    my($dbh, $campo) = @_ ;
    my $sth = $dbh->prepare("SELECT Categoria_id, $campo FROM
Categoria where $campo!='' and Activo=1");
    $sth->execute;

    return $sth;
}#end sub getCategorias

#Utiliza text2wave de Festival para contestarle al cliente.
sub decir {
    use File::Basename;
    use Digest::MD5 qw(md5_hex);

    my($text) = @_ ;
    my $hash = md5_hex($text);
    my $sounddir = "/var/lib/asterisk/sounds/tts";
    my $wavefile = "$sounddir/."tts-$hash.wav";
    my $t2wp= "/usr/bin/";

    unless (-f $wavefile) {
        open(fileOUT, ">$sounddir/."say-text-$hash.txt");
        print fileOUT "$text";
        close(fileOUT);
        my $execf=$t2wp."text2wave -F 8000 -o $wavefile
$sounddir/say-text-$hash.txt > /dev/null";
        system($execf);
        unlink($sounddir/."say-text-$hash.txt");
    }
}

```

```

        $AGI->stream_file('tts/' . basename($wavfile, ".wav"));
        unlink($wavfile);
    }#end sub decir

# Retorna el número de líneas existentes en el archivo temporal.
sub getNumeroLineasTmp{
    my($filename) = @_ ;

    open(MYINPUTFILE, "</tmp/$filename"); # open for input
    my(@lines) = <MYINPUTFILE>; # read file into list
    @lines = sort(@lines); # sort the list
    my $num_lineas=@lines;
    close(MYINPUTFILE);

    return $num_lineas;
}#end sub getNumeroLineasTmp

# Se menciona cada categoría activa y luego si no se ha elegido a
ninguna se ofrece la opción de ir al menú de categorías. Si no
responde, se redirecciona al menú principal
sub presentarCategorias {
    my($res, $max_tries, $max_time_record, $beep_file,
    $max_confirm_tries) = @_ ;

    decir("Diga okay o sí luego de escuchar la categoría para
acceder a sus ítems.");
    decir("Diga Cancelar para regresar al menú principal. ");
    decir("Diga Chao para cerrar la llamada. ");

    while (my @row = $res->fetchrow_array ){
        decir("@row[1]");          #decir el nombre de la categoría

        my $tries=0;
        while ($tries < $max_tries) {
            my $vresponse = '';
            $AGI->stream_file($beep_file, '');
            $AGI-
>record_file("/tmp/$$, 'wav', '0', $max_time_record);
            $vresponse = asr("/tmp/$$.wav");
            $AGI->verbose("# Intento elegir categoría: $tries,
vresponse: $vresponse");

            if ($vresponse =~ /CHAO/i) {
                $AGI->exec("AGI chao.agi");
                return undef;
            }elsif ($vresponse =~ /CANCELAR|CANCELLAR/i){
                $AGI->verbose("Va al Menu principal");
                $AGI->exec("AGI menu_principal.agi");
                return undef;
            }
        }
    }
}

```

```

        }elseif ($vresponse =~
/SEE|SIH|SI|OK|OHK|OKAY|OHKAY/i) {
            $AGI->set_variable('selected_categoria_id',
"@row[0]");
            $AGI->exec("AGI select_item.agi");
            return undef;
        }
        $tries++;
    } #end while intentos
} #end while categorias

#Si existe el file temporal de items seleccionados por confirmar
y tiene lineas
my $tmp_file_noconf=$AGI->get_variable('tmp_file_noconf');
my $lineas_items=0;
$lineas_items=getNumeroLineasTmp($tmp_file_noconf);

if($lineas_items>0){          #Tiene lpedido por confirmar, va a
revisar Pedido
    $AGI->verbose("El tmp tiene $lineas_items lineas.");
    $AGI->exec("AGI confirmar_pedido.agi");
    return undef;
}else{          #TerminÃ³ el listado de categorÃas y no ha
seleccionado Ãtems..
    decir("No seleccionoo ninguna categoria.");
    $AGI->verbose("No seleccionoo ninguna categoria.");
    decir("Diga okey o Sii para repetir el Menuu de Categoriias
o diga Cancelar o noo para volver al Menuu Principal");

    my $tries=0;
    while ($tries < $max_confirm_tries) {
        my $vresponse = '';
        $AGI->stream_file($beep_file,'');
        $AGI-
>record_file("/tmp/$$", 'wav', '0', $max_time_record);
        $vresponse = asr("/tmp/$$wav");

        $AGI->verbose("# Intento repetirCategorias: $tries,
vresponse: $vresponse");

        if ($vresponse =~ /CHAO/i) {
            $AGI->exec("AGI chao.agi");
            return undef;
        }elseif ($vresponse =~
/CANCELAR|NO|NOH|NOP|CANCEL|CANCEHLAR/i) {
            $AGI->verbose("Va al Menu principal");
            $AGI->exec("AGI menu_principal.agi");
            return undef;
        }
    }
}

```

```

        }elseif ($vresponse =~
/SEE|SIH|SI|OK|OHK|OKAY|OHKAY/i){ #Repite el menÃº de categorÃas
        $AGI->set_context('asr');
        $AGI->set_extension('104');
        $AGI->set_priority(1);
        return 1;
        }
        $tries++;
    } #end while intentos

    decir("Regresaraa al menu principal");
    $AGI->verbose("Va al Menu principal");
    $AGI->exec("AGI menu_principal.agi");
    return undef;
}
} #end sub

$AGI->verbose("Llega a select_categoria");
$dbh=conectarBase();
if($dbh){
    my $campo="Descripcion";

    my $num_categorias=getCuantasCategorias($dbh);
    $AGI->verbose("Cuantas cats: $num_categorias");

    if($num_categorias>0){
        my $res_categorias=getCategorias($dbh, $campo);

        my $max_tries=2;          #para decir ok al listar las
categorias
        my $max_time_record=3000;
        my $beep_file='beep';
        my $max_confirm_tries=2;
        my
$cat_seleccionada_id=presentarCategorias($res_categorias, $max_tries,
$max_time_record, $beep_file, $max_confirm_tries);
    }else{
        $AGI->verbose("Al momento no hay categoriias activas.");
        decir("Al momento no tenemos categoriias activas.");
        $AGI->verbose("Va al Menu principal");
        $AGI->exec("AGI menu_principal.agi");
        return undef;
    }
}
} #Error conexion
decir("Ocurrioo un error. SerÃ¡ redireccionado a un operador.");
$AGI->verbose("Va a hablar con un operador.");

$AGI->set_context('asr');

```

```

    $AGI->set_extension('789');
    $AGI->set_priority(1);
    return undef;
}

```

Select_item.agi

```

#!/bin/sh
exec perl -w -x $0
#!/perl
$|=1; #borrar buffer

use Asterisk::AGI;
my $AGI = new Asterisk::AGI;

%input = $AGI->ReadParse();

# Copyright (c) 2005 Josh McAllister
# This program is free software; you can redistribute it and/or modify
# it under the same terms as Perl itself.
sub decir {
    use File::Basename;
    use Digest::MD5 qw(md5_hex);

    my($text) = @_ ;
    my $hash = md5_hex($text);
    my $sounddir = "/var/lib/asterisk/sounds/tts";
    my $wavefile = "$sounddir/". "tts-$hash.wav";
    my $t2wp= "/usr/bin/";

    unless (-f $wavefile) {
        open(fileOUT, ">$sounddir."/say-text-$hash.txt");
        print fileOUT "$text";
        close(fileOUT);
        my $execf=$t2wp."text2wave -F 8000 -o $wavefile
$sounddir/say-text-$hash.txt > /dev/null";
        system($execf);
        unlink($sounddir."/say-text-$hash.txt");
    }
    # $AGI->stream_file('tts/'.basename($wavefile, ".wav"));
    $AGI->control_stream_file('tts/'.basename($wavefile, ".wav"));
    unlink($wavefile);
}#end sub decir

# Crea una conexi3n a la base de datos. Retorna un DataBaseHandle que
# maneja la conexi3n a MySQL
sub conectarBase {

```



```

        use strict;
        use DBI;
        my $dbh = DBI->connect(          # DataBase Handle que maneja la
conexiÃ³n a MySQL
            'DBI:mysql:asterisk:localhost:3306',
            'root',
            'root',
            {RaiseError => 1}
        ) or die "No se pudo conectar a MySQL: $DBI::errstr\n";

        return $dbh;
    }#end sub conectarBase

# Retorna el total de Ãtems activos correspondientes a la categorÃa
seleccionada.
sub getCuantosItemsXCategoria {
    my($categoria_id, $dbh, $campo) = @_ ;
    my $sth = $dbh->prepare("SELECT count(*) FROM Item where
$campo!='' and Activo=1 and Categoria_id=$categoria_id");
    $sth->execute;

    return $sth;
}#end sub getCuantosItemsXCategoria

sub getItemsXCategoria {
    my($categoria_id, $dbh, $campo) = @_ ;
    my $sth = $dbh->prepare("SELECT Item_id, $campo, Precio FROM Item
where $campo!='' and Activo=1 and Categoria_id=$categoria_id");
    $sth->execute;

    return $sth;
}#end sub getItemsXCategoria

#Escribe el texto en el archivo en el directorio especificado.
sub writeFile{
    my ($texto,$dir,$filename) = @_ ;
    my $var='>>'.$dir."".$filename;
    open (MYFILE,$var);
    print MYFILE "$texto\n";
    close MYFILE;
}#end sub writeFile

# Revisa los enteros y decimales en la variable tipo Decimal, para leer
los primeros como dÃ³lares y los otros como centavos.
sub decirPrecio{
    my ($precio_string, $que) = @_ ;

    my @precio = split(/\./, $precio_string);
    my $str_entero="";

```

```

my $str_decimal="";

if($precio[0]==1){
    $str_entero="1 doolar";
}elsif($precio[0]>1){
    $str_entero="$precio[0] doolares";
}

if($precio[1]==1){
    $str_decimal="1 centavo";
}elsif($precio[1]>1){
    $str_decimal="$precio[1] centavos";
}

# $AGI->verbose("En decirPrecio, str_entero:$str_entero,
str_decimal: $str_decimal, que: $que");
    decir("$que $str_entero $str_decimal");
}#end sub decirPrecio

# Dice nombre y precio de cada ítem, dando la opción de elegir la
cantidad de items con el teclado numérico.
sub presentarItemsParaSeleccionar {
    my($res,$beep_file,$wait_segundos,$max_items,$escape_key,$confirm
ar_key, $tmp_file_noconf) = @_;          #@: hashes, $:scalar

    decir("Al oír el ítem que desee, ingrese por teclado el número
de unidades. ");
    decir("El número máximo de items es");
    decir($max_items);
    decir("Presione $escape_key para regresar al menú de
categorías.");
    decir("Presione $confirmar_key para confirmar su pedido.");
    decir("Presione numeral o * para repetir el audio actual. ");

    while (my @row = $res->fetchrow_array ) {
        decirPrecio("@row[2]", "@row[1]");          #$precio_string,
$que

        my $max_digitos=2;
        my $tries=0;
        while ($tries < 1) {
            my $cantidad = $AGI->get_data($beep_file,
$wait_segundos, $max_digitos);

            if($cantidad!=""){
                if ($cantidad ==$confirmar_key){
                    my $tmp_file_noconf=$AGI-
>get_variable('tmp_file_noconf');

```

```

$AGI->verbose("tmp_file_noconf:
$tmp_file_noconf");

    if($tmp_file_noconf){
        #si se ha elegido algo va a confirmar el pedido
        $AGI->verbose("Va a Revisar
Pedido.");
        $AGI->exec("AGI
confirmar_pedido.agi");
        return undef;
    }else{
        $AGI->verbose("No puede ir a
Revisar Pedido.");
        decir("Al momento no ha
seleccionado ningun iitem.");
    }
    }elseif ($cantidad ==$escape_key){
        #Regresa al menu Principal
        $AGI->verbose("Presiono el codigo para
regresar al menu de categorias");
        $AGI->exec("AGI select_categoria.agi");
        return undef;
    }elseif($cantidad >0 && $cantidad <=$max_items){
        my $ha_elegido="";
        my $seleccion="";
        if($cantidad >1){
            $ha_elegido="Ha elegido $cantidad";
            $seleccion="@row[1]s";
        }else{
            $ha_elegido="Ha elegido $cantidad";
            $seleccion="@row[1]";
        }
        decir($ha_elegido." ".$seleccion);

        $AGI->set_variable('tmp_file_noconf',
$tmp_file_noconf);

        my $item_id="@row[0]";
        my $texto_linea=$item_id." ".$cantidad;
        $AGI->verbose("texto_linea:
$texto_linea"); #item_id, cantidad
        my
        $escribio=writeFile($texto_linea,"/tmp/", $tmp_file_noconf);
    }
    }
    $tries++;
} #end while intentos
} #end while items

```

```

        decir("Regresa al menuu de categoriias.");
        $AGI->verbose("Va al Menu de Categorias.");
        $AGI->exec("AGI select_categoria.agi");
        return undef;
    } #end sub presentarItemsParaSeleccionar

#Narra nombre y precio de cada item.
sub listarItems{
    my($res,$beep_file,$wait_segundos,$max_items,$escape_key,$confirm
ar_key, $tmp_file_noconf) = @_;          #@: hashes, $:scalar

    while (my @row = $res->fetchrow_array ) {
        decirPrecio("@row[2]", "@row[1]");          #$precio_string,
$que
        sleep 1;
    } #end while items

    decir("Regresa al menuu de categoriias.");
    $AGI->verbose("Va al Menu de Categorias.");
    $AGI->exec("AGI select_categoria.agi");
    return undef;
} #end sub listarItems

my $categoria_id=$AGI->get_variable('selected_categoria_id');
if ($categoria_id>0) {
    $dbh=conectarBase();
    if($dbh){
        my $campo="Descripcion";
        my $res_num_items=getCuantosItemsXCategoria($categoria_id,
$dbh, $campo);
        my $num_items=$res_num_items->fetchrow_array;

        if ($num_items>0) {
            my $res=getItemsXCategoria($categoria_id, $dbh,
$campo);

            my $wait_segundos=3000;
            my $beep_file='beep';
            my $max_items=15;
            my $escape_key=99;
            my $confirmar_key=98;
            my $tmp_file_noconf="data_noconf.txt";

            my $cedula=$AGI->get_variable('cedula');
            if($cedula==""){
                $AGI->verbose("CÃ©dula vacÃ­a, listo iitems");
                #
                decir("Para seleccionar iitems debe estar
autenticado. Diga cancelar despuÃ©s del tono para regresar al menuu de
categoriias.");
            }
        }
    }
}

```


Confirmar_pedido.agi

```
#!/bin/sh
exec perl -w -x $0
#!/perl
$|=1; #borrar buffer

use Asterisk::AGI;
my $AGI = new Asterisk::AGI;

%input = $AGI->ReadParse();

# Copyright (c) 2005 Josh McAllister
# This program is free software; you can redistribute it and/or modify
it under the same terms as Perl itself.
sub asr {
    use IO::Socket;
    use FileHandle;
    use IPC::Open2;

    my $file = shift or return undef;
    my $host = shift || 'localhost';
    my $port = shift || '1069'; # puerto del server
    my $fh;

    my $remote = IO::Socket::INET->new(
        Proto => "tcp",
        PeerAddr => "$host",
        PeerPort => "$port",
    ) or return undef;

    #Idea here being that you can pass a reference to an existing
file handle... not yet implemented, just pass a filename.
    if (ref $file) {
        my $fh = $file;
    } else {
        open (FH, $file) || return undef;
        $fh = *FH;
    }

    $file =~ /(gsm|wav)$/;
    my $type = $1;
    if ($type !~ /gsm|wav/) {
        warn "Unknown file type ($file)";
        return undef;
    }

    #print "FTYPE: $type\n";
}
```

```
$pid = open2(*SOXIN, *SOXOUT, "sox -t $type - -s -r 8000 -w -t  
wav - 2>/dev/null") || warn ("Could not open2.\n");
```

```
binmode $fh;  
binmode SOXIN;  
binmode SOXOUT;  
binmode $remote;
```

```
while (defined(my $b = read $fh, my($buf), 4096)) {  
    last if $b == 0;  
    $count += $b;  
    print SOXOUT $buf;  
}  
close SOXOUT;
```

```
$count = 0;  
my $sox = undef;  
while (defined(my $b = read SOXIN, my($buf), 4096)) {  
    last if $b == 0;  
    $count += $b;  
    $sox .= $buf;  
}  
print $remote length($sox) . "\n";  
print $remote "$sox";  
close SOXIN;
```

```
#print "DEBUG: Waiting for result.\n";
```

```
$count=0;  
while (defined(my $b = read $remote, my($buf), 4096)) {  
    last if $b == 0;  
    $count += $b;  
    $result .= $buf;  
}  
}
```

```
close $fh;  
close $remote;
```

```
return "$result";
```

```
}#end sub asr
```

```
# Crea una conexi3n a la base de datos. Retorna un DataBaseHandle que  
maneja la conexi3n a MySQL
```

```
sub conectarBase {  
    use strict;  
    use DBI;  
    my $dbh = DBI->connect(  
        'DBI:mysql:asterisk:localhost:3306',  
        'root',
```

```

        'root',
        {RaiseError => 1}
    ) or die "No se pudo conectar a MySQL: $DBI::errstr\n";

    return $dbh;
}#end sub conectarBase

#Utiliza text2wave de Festival para contestarle al cliente.
sub decir {
    use File::Basename;
    use Digest::MD5 qw(md5_hex);

    my($text) = @_ ;
    my $hash = md5_hex($text);
    my $sounddir = "/var/lib/asterisk/sounds/tts";
    my $wavefile = "$sounddir/."tts-$hash.wav";
    my $t2wp= "/usr/bin/";

    unless (-f $wavefile) {
        open(fileOUT, ">$sounddir."/say-text-$hash.txt");
        print fileOUT "$text";
        close(fileOUT);
        my $execf=$t2wp."text2wave -F 8000 -o $wavefile
$sounddir/say-text-$hash.txt > /dev/null";
        system($execf);
        unlink($sounddir."/say-text-$hash.txt");
    }
    $AGI->stream_file('tts/' . basename($wavefile, ".wav"));
    unlink($wavefile);
}#end sub decir

# Retorna un arreglo con el id, campo a leer (Descripcion), y precio del
item especificado.
sub getItemXItemId{
    my($dbh,$item_id,$campo) = @_ ;
    $query="SELECT Item_id, $campo, Precio FROM Item where $campo!=''
and Activo=1 and Item_id=$item_id";
    my $sth = $dbh->prepare($query);
    $sth->execute;

    return $sth;
}#end sub getItemXItemId

# Inserta un registro en la tabla Pedido. A%ste todav%a no tiene los
precios de los detalles. En caso de crearse correctamente, retorna el
Id con el que se guard% al pedido.
sub crearPedido{
    my ($dbh, $nombre_pedido, $pedido_activo, $fecha_pedido,
$pedido_estado_id, $direccion_id, $cedula) = @_ ;

```



```

use Mysql;

my $host = "localhost";
my $database = "asterisk";
my $user = "root";
my $pw = "root";

my $connect = Mysql->connect($host, $database, $user, $pw);
$connect->selectdb($database);

my $query="insert into Pedido(Nombre, Activo, Fecha_Creacion,
Estado_Pedido_id, Direccion_id, Persona_id)";
$query.=" values('$nombre_pedido', $pedido_activo, $fecha_pedido,
$pedido_estado_id, $direccion_id, (select Persona_id from Persona where
cedula=$cedula));";
my $execute = $connect->query($query);
my $lastid = $execute->insertid($query);

return $lastid;
}#end sub crearPedido

# Inserta un registro en la tabla Pedido_Detalle especificandole el Id
del Pedido padre.
sub crearPedidoDetalle{
    my($dbh, $pedido_id, $item_id, $cantidad, $iva,
$pedido_detalle_activo) = @_;

    my $query = "insert into Pedido_Detalle(Pedido_id, Nombre,
Activo, Item_id, Unidades, Precio_unidad, Precio_total, Precio_iva,
Precio_final)";
    $query .= " values($pedido_id, (select Nombre from Item where
Item_id =$item_id), $pedido_detalle_activo, $item_id, $cantidad,
(select Precio from Item where Item_id =$item_id), (select (select
Precio from Item where Item_id =$item_id)*$cantidad), (select (select
Precio from Item where Item_id =$item_id)*$cantidad*$iva), (select
(select Precio from Item where Item_id
=$item_id)*$cantidad*(1+$iva)));";

    my $sth = $dbh->prepare($query);
    $sth->execute;

    return $sth;
}#end sub crearPedidoDetalle

# Setea los precios de un pedido sumando los respectivos precios de sus
detalles.
sub actualizarPedido{
    my($dbh, $pedido_id) = @_;
```

```

        my $query = "update Pedido set Precio_total=(select
sum(Precio_total) from Pedido_Detalle where
Pedido_Detalle.Pedido_id=$pedido_id and Pedido_Detalle.Activo=1),
Precio_iva=(select sum(Precio_iva) from Pedido_Detalle where
Pedido_Detalle.Pedido_id=$pedido_id and Pedido_Detalle.Activo=1),
Precio_final=(select sum(Precio_final) from Pedido_Detalle where
Pedido_Detalle.Pedido_id=$pedido_id and Pedido_Detalle.Activo=1) where
Pedido.Pedido_id=$pedido_id;";

        my $sth = $dbh->prepare($query);
        $sth->execute;

    }#end sub actualizarPedido

# guardarPedido crea un pedido, recorre los detalles seleccionados
almacenados en el archivo temporal, crea un detalle en la base por cada
uno y luego actualiza los precios del pedido. Finalmente elimina el
archivo temporal y setea en '' la variable de canal tmp_file_noconf.
sub guardarPedido{
    my($dbh, $iva, $cedula, %items_confirmados) = @_;

    my $nombre_pedido="pruebaIvr";
    my $pedido_activo=1;
    my $fecha_pedido="NOW()";
    my $pedido_estado_id=2; #1=creado, 2=confirmado
    my $direccion_id=2;
    my $pedido_detalle_activo=1;

    my $pedido_id=crearPedido($dbh, $nombre_pedido, $pedido_activo,
$fecha_pedido, $pedido_estado_id, $direccion_id, $cedula);
    $AGI->verbose("pedido_id: $pedido_id");

    if($pedido_id>0){
        $AGI->verbose("items_confirmados: ");
        while (($item_id, $cantidad) = each(%items_confirmados)){
            $AGI->verbose("item_id: $item_id, cantidad:
$cantidad");
            crearPedidoDetalle($dbh, $pedido_id, $item_id,
$cantidad, $iva, $pedido_detalle_activo);
        }

        actualizarPedido($dbh, $pedido_id);

        my $filename="data_noconf.txt";
        my $file_deleted=unlink("/tmp/$filename");
        $AGI->verbose("file_deleted: $file_deleted");
        $AGI->set_variable('tmp_file_noconf', "");
        $AGI->verbose("Se blanqueo variable del tmp");
    }
}

```

```

        return $pedido_id;
    }#end sub guardarPedido

# Revisa los enteros y decimales en la variable tipo Decimal, para leer
los primeros como dólares y los otros como centavos.
sub decirPrecio{
    my ($precio_string, $que) = @_;

    my @precio = split(/\./, $precio_string);
    my $str_entero="";
    my $str_decimal="";

    if($precio[0]==1){
        $str_entero="1 dolar";
    }elsif($precio[0]>1){
        $str_entero="$precio[0] dolares";
    }

    if($precio[1]==1){
        $str_decimal="1 centavo";
    }elsif($precio[1]>1){
        $str_decimal="$precio[1] centavos";
    }

#     $AGI->verbose("En decirPrecio, str_entero:$str_entero,
str_decimal: $str_decimal, que: $que");
    decir("$que $str_entero $str_decimal");
}#end sub decirPrecio

#Dice el Nombre y precio base de cada detalle
sub getDecirDetallePrecio{
    my($unidades, $detalle_nombre, $precio_unidad) = @_;

    my $detalle="";
    if($unidades>0){
        $detalle="$unidades $detalle_nombre ";
    }
    else{
        $detalle="$unidades $detalle_nombre"."s
";}

    my $detalle_precio=($unidades * $precio_unidad);
    decirPrecio($detalle_precio, $detalle);

    return $detalle_precio;
}#end sub getDecirDetallePrecio

# Recorre cada línea del archivo temporal y leyendo la cantidad, nombre
y precio de cada detalle. Si el cliente no ignora un detalle, éste se
agregará al arreglo de detalles por guardar. Si el cliente confirma el
pedido, se guardara un pedido con los detalles de este arreglo.
sub revisarItems{

```

```

my($dbh,$filename, $campo, $line_delim, $max_confirm_tries,
$beep_file, $max_time_record, $iva, $cedula) = @_;

open(MYINPUTFILE, "</tmp/$filename"); # open for input
my(@lines) = <MYINPUTFILE>; # read file into list
@lines = sort(@lines); # sort the list

my $pedido_revisado=0;

while ($pedido_revisado==0) {
    my %items_confirmados = (); #hash para acumular los items
con sus cantidades
    my $precio_base_pedido=0;

    foreach $line (@lines){
        chomp($line);
        if($line!=""){
            my @par = split(/$line_delim/, $line);
            my $what=getItemXItemId($dbh,$par[0],$campo);
            my @item = $what->fetchrow_array;

            my $detalle_precio=($par[1]*$item[2]);
            decirPrecio($detalle_precio, "$par[1]
@item[1]"); # $precio_string, $que
            $AGI->verbose("item_id: $par[0], cantidad:
$par[1], detalle_precio: $detalle_precio");

            my $negado=0;
            my $tries=0;
            while ($tries < $max_confirm_tries &&
$negado==0) {

                my $vresponse = '';
                $vresponse = "";
                $AGI->stream_file($beep_file, '');
                $AGI-
>record_file("/tmp/$$", 'wav', '0', $max_time_record);
                $vresponse = asr("/tmp/$$.wav");

                my $ultima=$vresponse;
                $ultima =~ s/^..*?(\w+)[.!]??\s*$/\1/;

                #ultima del asr

                if($ultima=="ZEE"){
                    my @arr = split(/ /, $vresponse);
                    my $tam=scalar (@arr);
                    $ultima=$arr[$tam-2];
                }
                $AGI->verbose("# Intento negar detalle:
$tries, ultima: $ultima");

```

```

        if ($ultima eq "CHAO") {
            $AGI->exec("AGI chao.agi");
            return undef;
        }elseif ($ultima eq "CANCELAR" || $ultima
eq "CANCEHLAR") {
            decir("canceloo. regresar al menu
anterior. ");
        }elseif ($ultima eq "NO" || $ultima eq
"NOH" || $ultima eq "NOP") {
            #
            tmp
            que esta lienea ya no este en el
            $negado=1;
            decir("Se ignoraraa esta lienea en
el pedido.");
        }
        $tries++;
    } #end while intentos

    if($negado==0){
        #no negocio, acumular item y cantidad en el
hash items_confirmados. +precio_base del pedido
        $AGI->verbose("Agregar al hash el item_id
$par[0], $par[1] unidades");
        if($items_confirmados{$par[0]}){
            $cantidad_anterior=$items_confirmados{$par[0]};
            $AGI->verbose("Ya habia
$cantidad_anterior unidades del item_id $par[0]");
            $items_confirmados{$par[0]} =
$par[1]+$cantidad_anterior;
            $AGI->verbose("Ahora son
".$items_confirmados{$par[0]}." unidades del item_id $par[0]");
        }else{
            $AGI->verbose("No existia el
item_id $par[0]");
            $items_confirmados{$par[0]} =
$par[1];
            # hash, using variables
        }
        $precio_base_pedido+=$detalle_precio;
    }
}
    $vresponse = "";
    $negado=0;
} #end foreach linea
close(MYINPUTFILE);

#
    $AGI->verbose("precio_base_pedido: $precio_base_pedido,
num_items_confirmados: $num_items_confirmados");

```

```

        $AGI->verbose("precio_base_pedido: $precio_base_pedido,
num_items_confirmados: buscar");
        if($precio_base_pedido>0){
            $precio_final_pedido=$precio_base_pedido*(1+$iva);
            $AGI->verbose("precio_final_pedido:
$precio_final_pedido");
            decir("El precio base de su pedido seria");
            decirPrecio($precio_base_pedido, "");
            decir("El precio final de su pedido seria");
            decirPrecio($precio_final_pedido, "");
            decir("Diga Okey o Si para confirmar el pedido.");
        }else{
            decir("No han quedado items para realizar el
pedido.");
        }

        decir("Diga No para volver a revisar los iitems
seleccionados o Cancelar para regresar al menuu anterior. ");

        my $tries=0;
        while ($tries < $max_confirm_tries){

            my $vresponse = '';
            $AGI->stream_file($beep_file, '');
            $AGI-
>record_file("/tmp/$$", 'wav', '0', $max_time_record);
            $vresponse = asr("/tmp/$$wav");

            my $ultima=$vresponse;
            $ultima =~ s/^\.*?(\w+)[.!]??\s*/$1/;      #ultima del
asr

            if($ultima=="ZEE"){
                my @arr = split(/ /, $vresponse);
                my $tam=scalar (@arr);
                $ultima=$arr[$tam-2];
            }
            $AGI->verbose("#Intento confirmar pedido: $tries,
ultima palabra: $ultima");

            if ($ultima eq "CHAO") {
                $AGI->exec("AGI chao.agi");
                $pedido_revisado=1;
                return undef;
            }elseif ($ultima eq "CANCELAR" || $ultima eq
"CANCEHLAR") {

                decir("Regresaraa al menu principal");
                $AGI->verbose("Va al Menu principal");
                $AGI->exec("AGI menu_principal.agi");
                return undef;
            }
        }
    }
}

```

```

        }elseif ($ultima eq "NO" || $ultima eq "NOH" ||
$ultima eq "NOP") {
            #no cambio $pedido_revisado
            $AGI->verbose("Se repetiraa la revision de
items seleccionados. ");
        }elseif ($ultima eq "SEE" || $ultima eq "SIH" ||
$ultima eq "SI" || $ultima eq "OK" || $ultima eq "OHK" || $ultima eq
"OKAY" || $ultima eq "OHKAY") {
            if($precio_base_pedido>0){
                $AGI->verbose("Va a guardar el pedido.");
                my $pedido_id=guardarPedido($dbh, $iva,
$cedula, %items_confirmados);
                sleep 2;
                decir("Pedido registrado con el aidiid
$pedido_id");
                decir("Regresaraa al menu principal");
                $AGI->verbose("Va al Menu principal");
                $AGI->exec("AGI menu_principal.agi");
                return undef;
            }else{
                decir("No tiene items seleccionados
para crear un pedido.");}
        }
        $tries++;
    } #end while intentos
} #end while pedido_revisado==0

#Ha ingresado un pedido
decir("Regresaraa al menu principal");
$AGI->verbose("Va al Menu principal");
$AGI->exec("AGI menu_principal.agi");
return undef;
}#end sub revisarItems

my $cedula=$AGI->get_variable('cedula');
my $tmp_file_noconf=$AGI->get_variable('tmp_file_noconf');
my $length_cedula=length($cedula);
$AGI->verbose("En Confirmar Pedido, cedula: $cedula, tmp_file_noconf:
$tmp_file_noconf, length_cedula: $length_cedula");

if( ($length_cedula==10) && ($tmp_file_noconf gt '') ){
    decir("Confirmar pedido.");
    decir("Diga No al escuchar un iitem si desea eliminarlo.");
    decir("Diga Cancelar para regresar al menuu anterior.");
    #cual?

    $dbh=conectarBase();
    if($dbh){

```



```

my $sounddir = "/var/lib/asterisk/sounds/tts";
my $wavefile = "$sounddir/". "tts-$hash.wav";
my $t2wp= "/usr/bin/";

unless (-f $wavefile) {
    open(fileOUT, ">$sounddir."/say-text-$hash.txt");
    print fileOUT "$text";
    close(fileOUT);
    my $execf=$t2wp."text2wave -F 8000 -o $wavefile
$sounddir/say-text-$hash.txt > /dev/null";
    system($execf);
    unlink($sounddir."/say-text-$hash.txt");
}
# $AGI->stream_file('tts/'.basename($wavefile, ".wav"));
$AGI->control_stream_file('tts/'.basename($wavefile, ".wav"));
unlink($wavefile);
}#end sub decir

# Crea una conexi3n a la base de datos. Retorna un DataBaseHandle que
maneja la conexi3n a MySQL
sub conectarBase {
    use strict;
    use DBI;
    my $dbh = DBI->connect(
        'DBI:mysql:asterisk:localhost:3306',
        'root',
        'root',
        {RaiseError => 1}
    ) or die "No se pudo conectar a MySQL: $DBI::errstr\n";

    return $dbh;
}#end sub conectarBase

#Retorna el n3mero de detalles activos que tiene el pedido
especificado.
sub getCuantosDetallesXPedidoId{
    my($pedido_id, $dbh) = @_ ;
    my $query="SELECT count(*) FROM Pedido_Detalle where Activo=1 and
Pedido_id=$pedido_id;";

    my $sth = $dbh->prepare($query);
    $sth->execute;
    my $arr = $sth->fetchrow_array;
    return $arr;
}#end sub getCuantosDetallesXPedidoId

#Retorna los detalles activos correspondientes al pedido especificado.
sub getDetallesXPedidoId{
    my($pedido_id, $dbh) = @_ ;

```

```

        $query="SELECT i.Descripcion, pd.Unidades, pd.Precio_unidad,
pd.Precio_total, pd.Precio_iva, pd.Precio_final";
        $query.=" from Pedido_Detalle pd join Item i on
pd.Item_id=i.Item_id where pd.Pedido_id=$pedido_id and pd.Activo=1;";
        my $sth = $dbh->prepare($query);
        $sth->execute;

        return $sth;
    }#end sub getDetallesXPedidoId

# Guarda el texto recibido en el archivo en el directorio especificado.
sub writeFile{
    my ($texto,$dir,$filename) = @_ ;
    my $var='>>'.$dir."".$filename;
    open (MYFILE,$var);
    print MYFILE "$texto\n";
    close MYFILE;
}#end sub writeFile

# Revisa los enteros y decimales en la variable tipo Decimal, para leer
los primeros como dÃ³lares y los otros como centavos.
sub decirPrecio{
    my ($precio_string, $que) = @_ ;

    my @precio = split(/\./, $precio_string);
    my $str_entero="";
    my $str_decimal="";

    if($precio[0]==1){
        $str_entero="1 doolar";
    }elsif($precio[0]>1){
        $str_entero="$precio[0] doolares";
    }

    if($precio[1]==1){
        $str_decimal="1 centavo";
    }elsif($precio[1]>1){
        $str_decimal="$precio[1] centavos";
    }

    # $AGI->verbose("En decirPrecio, str_entero:$str_entero,
str_decimal: $str_decimal, que: $que");
    decir("$que $str_entero $str_decimal");
}#end sub decirPrecio

#Narra nombre y precio de cada Detalle
sub listarPedidoDetalles{

```

```

my($res,$beep_file,$wait_segundos,$max_items,$escape_key,$confirm
ar_key, $tmp_file_noconf) = @_;

while (my @row = $res->fetchrow_array ) { #Descripcion      |
Unidades | Precio_unidad | Precio_total | Precio_iva | Precio_final |
      $AGI->verbose("row[5]:@row[5], row[1]: @row[1], row[0]:
@row[0]");
      decirPrecio("@row[5]", "@row[1] @row[0]");
      # $precio_string, $que
      sleep 1;
    } #end while items
    decir("Regresaraa al listado de pedidos pendientes.");
    $AGI->verbose("Va a Revisar estado del pedido.");
    $AGI->exec("AGI revisar_estado_pedidos.agi");
    return undef;
} #end sub listarPedidoDetalles

my $selected_pedido_id=$AGI->get_variable('selected_pedido_id');
if ($selected_pedido_id>0) {
    $dbh=conectarBase();
    if($dbh){
        my $cedula=$AGI->get_variable('cedula');
        if($cedula==""){
            $AGI->verbose("CÃ©dula vacÃ­a, no listo iitems porque
el usuario no estÃ¡ autenticado.");
            decir("Debe estar autenticado para escuchar los
iitems de este pedido.");
            #quehago?
        }else{
            my $total_detalle=0;

            $total_detalle=getCuantosDetallesXPedidoId($selected_pedido_id,
$dbh);
            $AGI->verbose("El pedido $selected_pedido_id tiene
$total_detalle detalles. ");

            if($total_detalle>0){
                my
$res=getDetallesXPedidoId($selected_pedido_id, $dbh);
                if($res){
                    my $wait_segundos=3000;
                    my $beep_file='beep';
                    my $max_items=15;
                    my $escape_key=99;
                    my $confirmar_key=98;
                    my $tmp_file_noconf="data_noconf.txt";

```



```

use Asterisk::AGI;
my $AGI = new Asterisk::AGI;

%input = $AGI->ReadParse();

# Copyright (c) 2005 Josh McAllister
# This program is free software; you can redistribute it and/or modify
it under the same terms as Perl itself.
sub asr {
    use IO::Socket;
    use FileHandle;
    use IPC::Open2;

    my $file = shift or return undef;
    my $host = shift || 'localhost';
    my $port = shift || '1069';    # puerto del server
    my $fh;

    my $remote = IO::Socket::INET->new(
        Proto    => "tcp",
        PeerAddr => "$host",
        PeerPort => "$port",
    ) or return undef;

    if (ref $file) {
        my $fh = $file;
    } else {
        open (FH, $file) || return undef;
        $fh = *FH;
    }

    $file =~ /(gsm|wav)$/;
    my $type = $1;
    if ($type !~ /gsm|wav/) {
        warn "Unknown file type ($file)";
        return undef;
    }

    #print "FTYPE: $type\n";
    $pid = open2(*SOXIN, *SOXOUT, "sox -t $type - -s -r 8000 -w -t
wav - 2>/dev/null") || warn ("Could not open2.\n");

    binmode $fh;
    binmode SOXIN;
    binmode SOXOUT;
    binmode $remote;

    while (defined(my $b = read $fh, my($buf), 4096)) {

```

```

        last if $b == 0;
        $count += $b;
        print SOXOUT $buf;
    }
close SOXOUT;

$count = 0;
my $sox = undef;
while (defined(my $b = read SOXIN, my($buf), 4096)) {
    last if $b == 0;
    $count += $b;
    $sox .= $buf;
}
print $remote length($sox) . "\n";
print $remote "$sox";
close SOXIN;

#print "DEBUG: Waiting for result.\n";

$count=0;
while (defined(my $b = read $remote, my($buf), 4096)) {
    last if $b == 0;
    $count += $b;
    $result .= $buf;
}

close $fh;
close $remote;

return "$result";
}#end sub asr

#Utiliza text2wave de Festival para contestarle al cliente.
sub decir {
    use File::Basename;
    use Digest::MD5 qw(md5_hex);

    my($text) = @_ ;
    my $hash = md5_hex($text);
    my $sounddir = "/var/lib/asterisk/sounds/tts";
    my $wavefile = "$sounddir/."tts-$hash.wav";
    my $t2wp= "/usr/bin/";

    unless (-f $wavefile) {
        open(fileOUT, ">$sounddir/./say-text-$hash.txt");
        print fileOUT "$text";
        close(fileOUT);
        my $execf=$t2wp."text2wave -F 8000 -o $wavefile
        $sounddir/say-text-$hash.txt > /dev/null";
    }
}

```

```

        system($execf);
        unlink($sounddir."/say-text-$hash.txt");
    }
    $AGI->stream_file('tts/'.basename($wavefile, ".wav"));
    unlink($wavefile);
}#end sub decir

sub getInformacionSecciones{
    my %secciones_informacion = ();
    $secciones_informacion{0} = "Direccion";
    $secciones_informacion{1} = "Horarios de atencion";
    $secciones_informacion{2} = "Politica de entrega";
    $secciones_informacion{3} = "Sobre nuestro i ve ere";
    return %secciones_informacion;
}

my $digitos=10;
my $max_tries=2;
my $max_time_record=3000;
my $beep_file='beep';
#my $wait_segundos=5000;

decir("Informacion sobre el restaurante Macvoip.");

my %secciones_informacion= getInformacionSecciones();
$AGI->verbose("Opciones Mas Informacion");
while (($seccion_id, $seccion_opcion) = each(%secciones_informacion)){
    $AGI->verbose("seccion_id: $seccion_id, seccion_opcion:
    $seccion_opcion");
    decir($seccion_opcion);

    if($seccion_id==0){
        decir("Estamos ubicados en la Ciudadela blabla, Avenida
        blabla, manzana 123, frente a blabla. ");
    }elseif($seccion_id==1){
        decir("El restaurante estaa abierto desde las 10 de la
        maniana hasta las 10 de la noche.");
        decir("El servicio de pedido telefonico estaa habilitado
        las 24 horas.");
    }elseif($seccion_id==2){
        decir("Su pedido seraa entregado en un maaximo de 30
        minutos partir del momento de confirmacioon. ");
        decir("Si usted realiza su pedido fuera del horario de
        atencion del restaurante, su pedido sera comenzado a partir de las 9 de
        la maniana y seraa entregado en un maaximo de 30 minutos");
    }elseif($seccion_id==3){
        decir("Esta aplicacion de pedido telefonico fue
        desarrollada por Eloiisa Orozco.");
    }
}

```

```

    }

    #Luego de narrar el contenido de una seccion, beep para opcion a
salir
my $vresponse = '';
$AGI->stream_file($beep_file, '');
$AGI->record_file("/tmp/$$", 'wav', '0', $max_time_record);
$vresponse = asr("/tmp/$$wav");
$AGI->verbose("Opcion salir. vresponse: $vresponse");

if ($vresponse =~ /CHAO/i) {
    $AGI->exec("AGI chao.agi");
    return undef;
}elsif ($vresponse =~ /CANCELAR|CANCEHLAR/i) {
    decir("Regresaraa al menuu principal");
    $AGI->verbose("Va al menu principal");
    $AGI->exec("AGI menu_principal.agi");
    return undef;
}elsif ($vresponse =~ /SEE|SIH|SI|OK|OHK|OKAY|OHKAY/i){
    $AGI->set_context('asr');
    $AGI->set_extension('102');
    $AGI->set_priority(1);
    return 1;
}
} #end while secciones de Informacion

#TerminÃ³ la narracion de secciones
decir("Diga Okey o sii para repetir la informacioon. ");
$AGI->verbose("Diga Cancelar o noo para volver al menuu principal.");
decir("Diga Cancelar o noo para volver al menuu principal.");
decir("Diga chao para terminar la llamada.");

$tries=0;
while ($tries < $max_tries){
    $vresponse = '';
    $AGI->stream_file($beep_file, '');
    $AGI->record_file("/tmp/$$", 'wav', '0', $max_time_record);
    $vresponse = asr("/tmp/$$wav");
    $AGI->verbose("# Intento noSeguirEn info: $tries, vresponse:
    $vresponse");

    if ($vresponse =~ /CHAO/i) {
        $AGI->exec("AGI chao.agi");
        return undef;
    }elsif ($vresponse =~ /CANCELAR|CANCEHLAR|CANCEL/i) {
        $AGI->verbose("Va al menu principal");
        $AGI->exec("AGI menu_principal.agi");
        return undef;
    }elsif ($vresponse =~ /SEE|SIH|SI|OK|OHK|OKAY|OHKAY/i){

```



```

                $AGI->set_context('asr');                #Vuelve a Mas
informacion
                $AGI->set_extension('102');
                $AGI->set_priority(1);
                return 1;
            }
            $tries++;
        } #end while intentos

$AGI->set_context('asr');                #Vuelve a Mas informacion
$AGI->set_extension('102');
$AGI->set_priority(1);
return 1;

```

chao.agi

```

#!/bin/sh
exec perl -w -x $0
#!/perl
$|=1; #borrar buffer

use Asterisk::AGI;
my $AGI = new Asterisk::AGI;

%input = $AGI->ReadParse();

#Utilizar text2wave de Festival para contestarle al cliente.
sub decir {
    use File::Basename;
    use Digest::MD5 qw(md5_hex);

    my($text) = @_;
    my $hash = md5_hex($text);
    my $sounddir = "/var/lib/asterisk/sounds/tts";
    my $wavefile = "$sounddir/."tts-$hash.wav";
    my $t2wp= "/usr/bin/";

    unless (-f $wavefile) {
        open(fileOUT, ">$sounddir."/say-text-$hash.txt");
        print fileOUT "$text";
        close(fileOUT);
        my $execf=$t2wp."text2wave -F 8000 -o $wavefile
$sounddir/say-text-$hash.txt > /dev/null";
        system($execf);
        unlink($sounddir."/say-text-$hash.txt");
    }
    $AGI->control_stream_file('tts/'.basename($wavefile, ".wav"));
    unlink($wavefile);
}

```

```

}#end sub decir

# Crea una conexión a la base de datos.
sub conectarBase {
    use strict;
    use DBI;
    my $dbh = DBI->connect(          # DataBase Handle que maneja la
conexión a MySQL
        'DBI:mysql:asterisk:localhost:3306',
        'root',
        'root',
        {RaiseError => 1}
    ) or die "No se pudo conectar a MySQL: $DBI::errstr\n";

    return $dbh;
}#end sub conectarBase

#Guarda un registro en la tabla Llamada_calificacion especificando la
cedula del usuario que vota en caso de estar autenticado.
sub guardarCalificacionLlamada{
    my($dbh, $calificacion, $cedula) = @_ ;
    my %chan_vars = $AGI->ReadParse();

    $AGI->verbose("En guardarCalificacionLlamada, calificacion:
$calificacion, cedula: $cedula, callerid: ".$chan_vars{'callerid'}.",
uniqueid: ".$chan_vars{'uniqueid'});
    my $query = "";
    if($cedula==''){
        $query ="insert into Llamada_calificacion(Llamada_id,
Calificacion) values('".$chan_vars{'uniqueid'}."', $calificacion);";
    }else{
        $query ="insert into Llamada_calificacion(Llamada_id,
Calificacion, Persona_id) values('".$chan_vars{'uniqueid'}."',
$calificacion, (select Persona_id from Persona where
cedula=$cedula));";
    }

    my $sth = $dbh->prepare($query);
    $sth->execute;
    return $sth;
}#end sub guardarCalificacionLlamada

my $selected_pedido_id=$AGI->get_variable('selected_pedido_id');
my $tmp_file_noconf=$AGI->get_variable('tmp_file_noconf');

if($tmp_file_noconf gt ''){
    $AGI->verbose("tmp_file_noconf: $tmp_file_noconf. borrar.");
}

```

```

        my $file_deleted=unlink("/tmp/$tmp_file_noconf");
        $AGI->verbose("file_deleted: $file_deleted");
        decir("Se eliminaron los items seleccionados.");
    }else{
        $AGI->verbose("No habia tmp_file_noconf");
    }

    my $beep_file='beep';
    my $wait_segundos=3500;
    my $max_digitos=2;
    decir("Si gusta, califique del 1 al 10 su satisfaccioon en esta
    llamada.");

    my $calificacion = $AGI->get_data($beep_file, $wait_segundos,
    $max_digitos);
    my $guardada = 0;

    my $texto_despedida="";

    if($calificacion>=0 && $calificacion<=10){
        my $dbh = conectarBase();
        if($dbh){
            my $cedula=$AGI->get_variable('cedula');
            my $length_cedula=length($cedula);
            if($length_cedula==10){
                $guardada=guardarCalificacionLlamada($dbh,
                $calificacion, $cedula);
            }else{
                $guardada=guardarCalificacionLlamada($dbh,
                $calificacion);
            }
        }else{
            decir("No se pudo guardar su calificacion.");
        }
    }

    if($guardada){
        $texto_despedida= "Gracias por darnos su opinion. Hasta pronto.";
    }else{
        $texto_despedida= "Gracias por llamar. ";
    }
    decir($texto_despedida);

    $AGI->verbose($texto_despedida." Hangup.");
    $AGI->hangup();

```

error.agi

```

#!/bin/sh
exec perl -w -x $0
#!/perl
$|=1; #borrar buffer

use Asterisk::AGI;
my $AGI = new Asterisk::AGI;

%input = $AGI->ReadParse();

#Utiliza text2wave de Festival para contestarle al cliente.
sub decir {
    use File::Basename;
    use Digest::MD5 qw(md5_hex);

    my($text) = @_ ;
    my $hash = md5_hex($text);
    my $sounddir = "/var/lib/asterisk/sounds/tts";
    my $wavefile = "$sounddir/."tts-$hash.wav";
    my $t2wp= "/usr/bin/";

    unless (-f $wavefile) {
        open(fileOUT, ">$sounddir."/say-text-$hash.txt");
        print fileOUT "$text";
        close(fileOUT);
        my $execf=$t2wp."text2wave -F 8000 -o $wavefile
$sounddir/say-text-$hash.txt > /dev/null";
        system($execf);
        unlink($sounddir."/say-text-$hash.txt");
    }
    $AGI->control_stream_file('tts/'.basename($wavefile, ".wav"));
    unlink($wavefile);
}#end sub decir

my $error_base=$AGI->get_variable('error_base');
decir("Ha ocurrido un error.");

if($error_base==1){
    decir("SerÃ; redireccionado a un operador.");
    $AGI->verbose("Va a hablar con un operador.");

    $AGI->set_context('asr');
    $AGI->set_extension('789');
    $AGI->set_priority(1);
    return undef;
}else{

```

```
decir('RegresarÃ; al MenÃ° Principal');  
$AGI->verbose("Va al menu principal");  
$AGI->exec("AGI menu_principal.agi");  
return undef;  
}
```

Anexo 6: Código de la aplicación web interna

6.1 Nombre: login.html

```
<html>
  <head>

  </head>

  </head>
  <body class="sansserif">
    <h1>Sistema de Toma de Pedidos</h1>
    <h3>Ingrese su Cedula y Contraseña</h3>
    <div id="dialog">
      <form action="index.php" method="post">
        <div id="cont1">
          <tr>
            <td>Cedula :</td>
            <td>Contraseña :</td>
          </tr>
        </div>
        <div id="cont2">
          <tr>
            <td><input maxlength="20" name="login"
size="20" type="text"/></td>
            <td><input maxlength="20" name="password"
size="20" type="password" /></td>
          </tr>
        </div>
          <div id="cont3"><input type="submit"
value="Ingresar"/></div></form>
        </div>
      </body>
    </html>
```

Nombre: index.php

```
<?php
//include("db.php");
include("funciones.php");
//require("include.php");

define('SMARTY_DIR', '/usr/local/lib/Smarty-3.0.8/libs/');
require_once(SMARTY_DIR . 'Smarty.class.php');
$smarty = new Smarty();
$smarty->template_dir = '/var/www/html/templates';

$estado = 0; //mensaje de error
$logout = 0; //Salir del sistema
$usuario = ""; //Nombre de Usuario
$pass = ""; //Contraseña
//abrirdb(); //Conexión a la base de datos.
    if (isset($_POST["login"]))
    {
        $cedula = $_POST["login"];
        //$smarty->assign('cedula', $cedula);
        //echo $cedula;
    }
    if (isset($_POST["password"]))
    {
        $pass = $_POST["password"];
        //echo $pass;
    }
$conexion= connect_db();
if($conexion){
    $user=getUser($cedula, $pass);
    if($user=='0'){
        $mensaje="Error...!!! Verifique Cedula y Contraseña";
        $smarty->assign('mensaje', $mensaje);
    }else{
        $mensaje="Bienvenido :";
        $smarty->assign('user', $user);
    }
}
$msjS="Consultar Pedidos Pendientes";
//$msjS2="Consultar Pedidos Pagados";
$msjN="Volver al Inicio";
$smarty->assign('msjS', $msjS);
$smarty->assign('msjN', $msjN);
$smarty->display('correcto.tpl');
```

?>

Nombre: Actualizacion_estados_Pedidos Pendientes.php

```
<?php
include("funciones.php");
define('SMARTY_DIR', '/usr/local/lib/Smarty-3.0.8/libs/');
require_once(SMARTY_DIR . 'Smarty.class.php');

$smarty = new Smarty();
$smarty->template_dir = '/var/www/html/templates';

/**Logica**/

$cambiar="";
$cambiar=$_GET['cambiar'];
$conexion= connect_db();
if($cambiar=="si"){

    $idPedido="";
    if($conexion){
        $idPedido=$_GET['id_pedido'];
        //print_r($_GET);
        //$idPedido='31';
        setEstadoPedido($idPedido);
    }
}
//else{

    if($conexion){
        $pedidos=getPedidos();
    }
    $totalMin=20;
    $arreglo=array();
    $i=1;
    foreach($pedidos as $pedido){
        foreach($pedido as $campo=>$y){
            if($campo=="Fecha_Creacion"){

                if($i%2=="0"){
                    $totalMin=getMin($y);
                    //echo $totalMin;
                    if($totalMin>45)
                        $color='#FF0000';//Color Rojo
                }
                else if($totalMin>30 &&
                    $totalMin<45)
                    $color='orange';
                else if($totalMin>15 &&
                    $totalMin<30)
                    $color='#FFFF00';//Color Amarillo
                else $color='#00FF00';//Color Verde
                $arreglo[]=$color;
            }
        }
    }
}
```



```

        }
        $i=$i+1;
    }
}
}
//}
$smarty->assign('titulo',"Sistema de Toma de Pedidos");
$smarty->assign('titulo2',"Actualizar Estados de Pedidos
Pendientes");
$smarty->assign('pedidos',$pedidos);
$smarty->assign('arreglo',$arreglo);
$smarty->display('consulta.tpl');

?>

```

Nombre: **consultarPersonas.php**

```

<?php
include("funciones.php");
define('SMARTY_DIR', '/usr/local/lib/Smarty-3.0.8/libs/');
require_once(SMARTY_DIR . 'Smarty.class.php');

$smarty = new Smarty();
$smarty->template_dir = '/var/www/html/templates';

/**Logica**/

$accion=$_GET['accion'];
$conexion= connect_db();

if($accion=="guardar"){
    //Actualiza
    $idP=$_GET['idPersona'];
    $name1=$_GET['nombre1'];
    $name2=$_GET['nombre2'];
    $lname1=$_GET['apellido1'];
    $lname2=$_GET['apellido2'];
    //print_r($_GET);
    if($conexion){
        setPersona($idP, $name1, $name2, $lname1,$lname2);
    }
}

```

```

    }

    if($conexion){
        $personas=getPersonas();
    }
    //print_r($personas);
    $smarty->assign('titulo',"Sistema de Toma de Pedidos");
    $smarty->assign('titulo2',"Listado de Clientes");
    $smarty->assign('personas',$personas);
    $smarty->display('consultPerson.tpl');

?>

```

Nombre: detalle_pedido.php

```

<?php
include("funciones.php");
define('SMARTY_DIR', '/usr/local/lib/Smarty-3.0.8/libs/');
require_once(SMARTY_DIR . 'Smarty.class.php');

$smarty = new Smarty();
$smarty->template_dir = '/var/www/html/templates';

/**Logica**/
    $conexion= connect_db();
    if($conexion){
        $pedidos=getDetalles_Pedidos($_GET['id_pedido']);
        $smarty->assign('pedidos',$pedidos);
    }
    $smarty->display('detail.tpl');
//Se utiliza la funcion para convertir las horas
?>

```

Nombre: consultarPedidosPagados.php

```

<?php
include("funciones.php");
define('SMARTY_DIR', '/usr/local/lib/Smarty-3.0.8/libs/');
require_once(SMARTY_DIR . 'Smarty.class.php');
$smarty = new Smarty();
$smarty->template_dir = '/var/www/html/templates';

```

```

/**Logica**/
$bandera="0";
$bandera=$_GET['bandera'];
$conexion= connect_db();

if($bandera=="1"){
    //Consulta x Fecha
    $date1=$_GET['date1'];
    $date2=$_GET['date2'];
    //echo "Entro en el if";
    if($conexion){
        $pedidos=getPedidosPagadosxFecha($date1,$date2);
    }
}
else{
    //Consulta los ultimos 20 Pedidos Pagos
    //echo "Entro en el else";
    if($conexion){
        $pedidos=getUltimosPedidosPagados();
    }
}

$smarty->assign('titulo',"Sistema de Toma de Pedidos");
$smarty->assign('titulo2',"Consulta de Pedidos Pagados");
$smarty->assign('pedidos',$pedidos);
$smarty->display('consultaP.tpl');

?>

```

Nombre: funciones.php

```

<?php
function connect_db(){
    $db_connection=0;
    $db_connection = mysql_connect('localhost','root') or die
("Error en al crear la conexion. ".mysql_error());
    if($db_connection){
        $db_select = mysql_select_db('test') or die ("Error al
seleccionar la base. ".mysql_error());
        $con_base=true;
    }
    return $db_connection;
}

//Fns a la base
function getPedidos(){

```

```

        $query = "SELECT p.Pedido_id, p.Fecha_Creacion, p.Precio_total,
p.Precio_iva, p.Precio_final, ep.Nombre as Estado, per.Nombre1,
per.Apellido1, c.Nombre as Ciudadela, d.Indicaciones FROM Pedido p JOIN
Estado_Pedido ep ON p.Estado_Pedido_id=ep.Estado_Pedido_id JOIN Persona
per ON p.Persona_id=per.Persona_id JOIN Direccion d ON
d.Direccion_id=p.Direccion_id JOIN Ciudadela c ON
d.Ciudadela_id=c.Ciudadela_id where p.Estado_Pedido_id!="6"."";
        $result = @mysql_query($query);

        if($result){
            $cats="";
            while ($row = mysql_fetch_array($result)) {
                $cats[]=$row;
            }
            return $cats;
        }
        return 0;
    }
function getDetalles_Pedidos($PedidoID){
    $query = "Select Nombre,
Activo,Unidades,Precio_unidad,Precio_iva, Precio_final from
Pedido_Detalle pd where Pedido_id='$PedidoID'";
    $result = @mysql_query($query);

    if($result){
        $cats="";
        while ($row = mysql_fetch_array($result)) {
            $cats[]=$row;
        }
        return $cats;
    }
    return 0;
}

function getMin($y){
    //Fecha Pedido.
    $Año="$y[0]".$y[1]".$y[2]".$y[3]";
    $mes="$y[5]".$y[6]";
    $dia= "$y[8]".$y[9]";
    $hora= "$y[11]".$y[12]";
    $min= "$y[14]".$y[15]";

    // Fecha Actual
    $x= date("Y-m-d H:i:s");
    //ECHO $x;
    $Año±ox="$x[0]".$x[1]".$x[2]".$x[3]";

```

```

$mesx="$x[5]".$x[6]";
$diax= "$x[8]".$x[9]";
$horax= "$x[11]".$x[12]";
$minx= "$x[14]".$x[15]";

//Diferencia
$daño=($año-$año)*525600;
$dmes=($mesx-$mes)*43200;
$ddia=($diax-$dia)*1440;
$dhora=($horax-$hora)*60;
$dmin=$minx-$min;
$totalMin=$daño+$dmes+$ddia+$dhora+$dmin;

return $totalMin;

}

function getUser($Cedula, $Pass){
    $query = "SELECT Nombre1, Apellido1, Password, Cedula FROM
Persona WHERE Cedula='$Cedula' and Password='$Pass'";
    //echo $Cedula."datos que entran a la funcion".$Pass;
    $result = @mysql_query($query);
    if($result){
        $cats="";
        while ($row = mysql_fetch_array($result)) {
            $cats[]=$row;
        }
        /*if($cats==""){
            echo "No sale nada";
        }
        print_r($cats);*/
        return $cats;
    }
    return 0;
}

function setEstadoPedido($idPedido){
    $query = "SELECT Estado_Pedido_id FROM Pedido WHERE
Pedido_id='$idPedido'";
    $result = @mysql_query($query);
    if($result){
        $cats="";
        while ($row = mysql_fetch_array($result)) {
            $cats[]=$row;
        }
    }
    $state=$cats[0]['Estado_Pedido_id'];
    $newState="";
}

```

```

        if($state>0 && $state<6 ){
            if($state=="1"){ $newState="2";}
            else if($state=="2"){ $newState="3";}
            else if($state=="3"){ $newState="4";}
            else if($state=="4"){ $newState="5";}
            else if($state=="5"){ $newState="6";}
            $query2 = "UPDATE Pedido SET Estado_Pedido_id='$newState'
WHERE Pedido_id='$idPedido'";
            $result2 = @mysql_query($query2);
        }
    }
}

```

```

function getUltimosPedidosPagados() {
    $query = "SELECT p.Pedido_id, p.Fecha_Creacion, p.Precio_total,
p.Precio_iva, p.Precio_final, ep.Nombre as Estado, per.Nombre1,
per.Apellido1, c.Nombre as Ciudadela, d.Indicaciones FROM Pedido p JOIN
Estado_Pedido ep ON p.Estado_Pedido_id=ep.Estado_Pedido_id JOIN Persona
per ON p.Persona_id=per.Persona_id JOIN Direccion d ON
d.Direccion_id=p.Direccion_id JOIN Ciudadela c ON
d.Ciudadela_id=c.Ciudadela_id where p.Estado_Pedido_id="."6"."";
    $result = @mysql_query($query);
    if($result){
        $cats="";
        while ($row = mysql_fetch_array($result)) {
            $cats[]=$row;
        }
        return $cats;
    }
    return 0;
}

```

```

function getPedidosPagadosxFecha($date1 , $date2){
    $query = "SELECT p.Pedido_id, p.Fecha_Creacion, p.Precio_total,
p.Precio_iva, p.Precio_final, ep.Nombre as Estado, per.Nombre1,
per.Apellido1, c.Nombre as Ciudadela, d.Indicaciones FROM Pedido p JOIN
Estado_Pedido ep ON p.Estado_Pedido_id=ep.Estado_Pedido_id JOIN Persona
per ON p.Persona_id=per.Persona_id JOIN Direccion d ON
d.Direccion_id=p.Direccion_id JOIN Ciudadela c ON
d.Ciudadela_id=c.Ciudadela_id where p.Estado_Pedido_id="."6"." and
p.Fecha_Creacion >= '$date1' and p.Fecha_Creacion <= '$date2'";
    $result = @mysql_query($query);

    if($result){
        $cats="";
        while ($row = mysql_fetch_array($result)) {

```

```

        $cats[]=$row;
    }
    return $cats;
}
return 0;
}

function getPersonas(){
    /*$query = "SELECT Persona_id,Cedula,Nombre1, Nombre2, Apellido1,
Apellido2, Fecha_Registro from Persona;";*/
    $query = "SELECT * from Persona;";
    $result = @mysql_query($query);

    if($result){
        $cats="";
        while ($row = mysql_fetch_array($result)) {
            $cats[]=$row;
        }
        return $cats;
    }
    return 0;
}

function getPersona($PedidoID){
    $query = "Select * from Persona where Persona_id='$PedidoID'";
    $result = @mysql_query($query);

    if($result){
        $cats="";
        while ($row = mysql_fetch_array($result)) {
            $cats[]=$row;
        }
        return $cats;
    }
    return 0;
}

```

```

function setPersona($idPersona, $nombre1, $nombre2,
$apellido1,$apellido2){

    $query = "UPDATE Persona SET Nombre1='$nombre1',
Nombre2='$nombre2', Apellido1='$apellido1', Apellido2='$apellido2'
WHERE Persona_id='$idPersona'";
    $result2 = @mysql_query($query);
}

?>

```

Nombre: **correcto.tpl**

```

<html>
    <head>

    </head>
    <body style="font-family:Arial">
        <h1>Sistema de Toma de Pedidos</h1>
        {section name=a loop=$user}
            <div id="frame1" >
                <h2>Bienvenido {$user[a].Nombre1}
{$user[a].Apellido1}</h2>
                <p><a href="consultarPedidos.php">{$msjS}</a></p>
                <p><a href="consultarPedidosPag.php">Consultar
Pedidos Pagados</a></p>
                <p><a href="consultarPersonas.php">Consultar
Clientes</a></p>
                <p><a href="login.html">Salir del Sistema</a></p>
            </div>
            {sectionelse}
                <div id="frame2">
                    <p>Verifique su Cedula y Contraseña</p>
                    <p><a href="login.html">{$msjN}</a></p>
                </div>
            {/section}

        </body>
</html>

```

Nombre: **consultarPedidosPag.tpl**

```

<html>

```



```

<head>

</head>
<body>
<h1 style="text-align:center">{$titulo}</h1>
<h2 style="text-align:center">{$titulo2}</h2>
<p><a href="index2.php">Regresar al MenÃº Principal </a></p>

<div>
<FORM ACTION="consultarPedidosPag.php" METHOD="GET">
<INPUT TYPE="hidden" NAME="bandera" VALUE="1"/>
<tr>
<td>Fecha Inicio</td>
<td><INPUT TYPE=TEXT NAME=date1 VALUE="aaaa-mm-
dd"></td>
</tr>
<tr>
<td>Fecha Final</td>
<td><INPUT TYPE=TEXT NAME=date2 VALUE="aaaa-mm-
dd"></td>
</tr>
<td><input type="submit" value="Consultar por Fecha"
/></td>
</FORM>
</div>

<div><table>
<tr>
<th>Pedido_id</th>
<th>Fecha_Creacion</th>
<th>Precio_total</th>
<th>Precio_iva</th>
<th>Precio_final</th>
<th>Estado</th>
<th>Nombre</th>
<th>Apellido</th>
<th>Ciudadela</th>
<th>Indicaciones</th>

</tr>
{section name=customer loop=$pedidos}

<tr
onclick="location.href='detalle_pedido.php?id_pedido={$pedidos[customer
].Pedido_id}'">
<td>{$pedidos[customer].Pedido_id}</td>
<td>{$pedidos[customer].Fecha_Creacion}</td>
<td>{$pedidos[customer].Precio_total}</td>

```

```

        <td>{$pedidos[customer].Precio_iva}</td>
        <td>{$pedidos[customer].Precio_final}</td>
        <td>{$pedidos[customer].Estado}</td>
        <td>{$pedidos[customer].Nombre1}</td>
        <td>{$pedidos[customer].Apellido1}</td>
        <td>{$pedidos[customer].Ciudadela}</td>
        <td>{$pedidos[customer].Indicaciones}</td>

    </tr>
    </section>
</table></div>
</body>
</html>

```

Nombre: **consulta.tpl**

```

<html>
  <head>
    <style type="text/css">
      table, td, th{
        border:1px solid #3C9DD0;}
      th{
        background-color:#64AAD0;
        color:white;}
      h1,h2{
        margin-top:3%;
        text-align:center;}
    </style>
  </head>
  <body style="font-family:Arial">
    <h1>{$titulo}</h1>
    <h2>{$titulo2}</h2>
    <p><a href="index2.php">Regresar al MenÃº Principal </a></p>

    <div><table>
      <tr>
        <th>Pedido_id</th>
        <th>Fecha_Creacion</th>
        <th>Precio_total</th>
        <th>Precio_iva</th>
        <th>Precio_final</th>
        <th>Estado</th>
        <th>Nombre</th>
        <th>Apellido</th>
        <th>Ciudadela</th>
        <th>Indicaciones</th>

```

```

<th></th>

</tr>
{assign var="col" value="0"}
{section name=customer loop=$pedidos}

<tr
onclick="location.href='detalle_pedido.php?id_pedido={$pedidos[customer
].Pedido_id}'" style="background-color:{$arreglo[$col]}">
<td>{$pedidos[customer].Pedido_id}</td>
<td>{$pedidos[customer].Fecha_Creacion}</td>
<td>{$pedidos[customer].Precio_total}</td>
<td>{$pedidos[customer].Precio_iva}</td>
<td>{$pedidos[customer].Precio_final}</td>
<td>{$pedidos[customer].Estado}</td>
<td>{$pedidos[customer].Nombrel}</td>
<td>{$pedidos[customer].Apellido1}</td>
<td>{$pedidos[customer].Ciudadela}</td>
<td>{$pedidos[customer].Indicaciones}</td>
{assign var="col" value=$col+1}

{if $pedidos[customer].Estado eq "Creado"}
{assign var="verbo" value="Confirmar"}
{elseif $pedidos[customer].Estado eq "Confirmado"}
{assign var="verbo" value="Comenzar"}
{elseif $pedidos[customer].Estado eq "Comenzado"}
{assign var="verbo" value="Despachar"}
{elseif $pedidos[customer].Estado eq "Despachado"}
{assign var="verbo" value="Entregar"}
{elseif $pedidos[customer].Estado eq "Entregado"}
{assign var="verbo" value="Pagar"}
{else}
{assign var="verbo" value="Crear"}
{/if}
<td><FORM ACTION="consultarPedidos.php" METHOD="GET">
<INPUT TYPE="hidden" NAME="id_pedido"
VALUE="{$pedidos[customer].Pedido_id}"/>
<INPUT TYPE="hidden" NAME="cambiar" VALUE="si"/>
<input type="submit" value="{ $verbo}" />
</FORM></td>

</tr>
{/section}
</table></div>
</body>
</html>

```

Nombre: **detail.tpl**

```

<html>
  <head>

  </head>
  <body style="font-family:Arial">
    <h1>Sistema de Toma de Pedidos</h1>
    <h2>Detalle del Pedido</h2>
    <div id="link"><a href="consultarPedidos.php">Regresar
Consultar Pedidos </a></div>
    <div id="tabla"><table >
      <tr>
        <th>Nombre</th>
        <th>Activo</th>
        <th>Unidades</th>
        <th>Precio_unidad</th>
        <th>Precio_iva</th>
        <th>Precio_final</th>
      </tr>

      {section name=customer loop=$pedidos}
      <tr>
        <td>{$pedidos[customer].Nombre}</td>
        <td>{$pedidos[customer].Activo}</td>
        <td>{$pedidos[customer].Unidades}</td>
        <td>{$pedidos[customer].Precio_unidad}</td>
        <td>{$pedidos[customer].Precio_iva}</td>
        <td>{$pedidos[customer].Precio_final}</td>
      </tr>
      {/section}
    </table></div>
  </body>
</html>

```

Nombre: actCliente.tlp

```

<html>
  <head>
    <style type="text/css">

      h1,h2{
        margin-top:3%;
        text-align:center;}
      table{
        margin-top:1%;
        margin-left:38%;
        position:absolute;

```

```

    }
    #guardar{
        margin-top:20%;
        margin-left:42%;
        position:absolute;
    }
    #cancelar{
        margin-top:-14%;
        margin-left:60%;
        position:relative;
    }
}

</style>

</head>
<body style="font-family:Arial">
    <h1>Sistema de Toma de Pedidos</h1>
    <h2>Actualizacion Datos de Clientes</h2>
    <div id="tabla">

        {section name=cust loop=$persona}
        <FORM ACTION="consultarPersonas.php" METHOD="GET">
            <INPUT TYPE="hidden" NAME="idPersona"
VALUE="{ $persona[cust].Persona_id}"/>
            <table>
                <tr>
                    <td>Nombre1: </td>
                    <td><INPUT TYPE=TEXT NAME=nombre1
VALUE="{ $persona[cust].Nombre1}"></td>
                </tr>
                <tr>
                    <td>Nombre2: </td>
                    <td><INPUT TYPE=TEXT NAME=nombre2
VALUE="{ $persona[cust].Nombre2}"></td>
                </tr>
                <tr>
                    <td>Apellido1: </td>
                    <td><INPUT TYPE=TEXT NAME=apellido1
VALUE="{ $persona[cust].Apellido1}"></td>
                </tr>
                <tr>
                    <td>Apellido2: </td>
                    <td><INPUT TYPE=TEXT NAME=apellido2
VALUE="{ $persona[cust].Apellido2}"></td>
                </tr>
            </table>
        </FORM>
    </div>

```

```

                <td>Direccion: </td>
                <td><INPUT TYPE=TEXT NAME=direccion
VALUE=""></td>
            </tr>
            <tr>
                <td>TelÃ©fono: </td>
                <td><INPUT TYPE=TEXT NAME=direccion
VALUE=""></td>
            </tr>
        </table>
        <INPUT TYPE="hidden" NAME="accion" VALUE="guardar"/>
        <div id="guardar"><input type="submit"
value="Guardar" /></td><div>
        </FORM>
        <FORM ACTION="consultarPersonas.php" METHOD="GET">
        <INPUT TYPE="hidden" NAME="accion"
VALUE="cancelar"/>
        <div id="cancelar"><input type="submit"
value="Cancelar" /></td><div>
        </FORM>
        {/section}
    </div>
</body>
</html>

```

Nombre: **consulPerson.tpl**

```

<html>
    <head>
        <style type="text/css">
            h1,h2{
                margin-top:3%;
                text-align:center;}
            #tabla{
                margin-left:20%;
                margin-top:2%;
                position:absolute;
                width:100%;
            }
            #link{

```

```

        margin-left:0%;
        margin-bottom:2%;
        position:relative;
        width:20%;
    }
</style>
</head>
<body style="font-family:Arial">
    <h1>{$titulo}</h1>
    <h2>{$titulo2}</h2>

    <div id="tabla"><table>
        <div id="link"><a href="index2.php">Regresar al MenÃº
Principal </a></div>
        <tr>
            <th>Cedula</th>
            <th>Nombre1</th>
            <th>Nombre2</th>
            <th>Apellido1</th>
            <th>Apellido2</th>
            <th>Fecha_Registro</th>
            <th></th>

        </tr>
        {section name=customer loop=$personas}

        <tr>
            <td>{$personas[customer].Cedula}</td>
            <td>{$personas[customer].Nombre1}</td>
            <td>{$personas[customer].Nombre2}</td>
            <td>{$personas[customer].Apellido1}</td>
            <td>{$personas[customer].Apellido2}</td>
            <td>{$personas[customer].Fecha_Registro}</td>
            <td><FORM ACTION="actualizarDatos.php" METHOD="GET">
                <INPUT TYPE="hidden" NAME="Persona_id"
VALUE="{$personas[customer].Persona_id}"/>
                <INPUT TYPE="hidden" NAME="cambiar" VALUE="si"/>
                <input type="submit" value="Actualizar" />
            </FORM></td>

        </tr>
        {/section}
    </table></div>

</body>
</html>

```

REFERENCIAS BIBLIOGRÁFICAS

- [1] Estrategiamagazine - “El Servicio al Cliente como Diferenciación” - Disertación de Jacques Horovitz en el marco del Programa Integral Hacia la Calidad Total organizado en Buenos Aires por la Fundación OSDE, <http://www.estrategiamagazine.com/marketing/el-servicio-al-cliente-como-diferenciacion-calidad-total-estrategias-reducir-costos-satisfaccion-cliente> , febrero 2003
- [2] The Centre for Speech Technology Research - University of Edinburgh - The Festival Speech Synthesis System, <http://www.cstr.ed.ac.uk/projects/festival/> , fecha de consulta: 1 junio 2011
- [3] Sheets Kris, Terence Jimmy ,García Marcelo y Saavedra Dany , “Asterisk en Español”, http://itaki.net/espanol/asterisk_espanol.pdf , fecha de consulta: 27 julio 2011, Pág. 4
- [4] Universidad de Sevilla, “Asterisk”, <http://bibing.us.es/proyectos/abreproy/11379/fichero/memoria%252F4.pdf> , fecha de consulta: 28 julio 2011, Pág. 11

- [5] Jiménez Ramón, "Teoría de Contextos de Asterisk – Parte I",
<http://blog.manuelviera.es/2011/01/ficheros-de-configuracion-y-directorios-en-asterisk/> , fecha de consulta: 26 julio 2011
- [6] Viera Manuel, "Directorios de Asterisk",
<http://blog.manuelviera.es/2011/01/ficheros-de-configuracion-y-directorios-en-asterisk/> , fecha de consulta: agosto 2011
- [7] Voip-Info, "Asterisk AGI", <http://www.voip-info.org/wiki/view/Asterisk+AGI> ,
fecha de consulta: 28 julio 2011.
- [8] Voxeo, "What is VoiceXML", <http://www.voxeo.com/library/voicexml.jsp> , fecha
de consulta: 10 julio 2011.
- [9] W3C-Recommendation, "Voice Extensible Markup Language",
<http://www.w3.org/TR/voicexml20/> , fecha de consulta: 2 agosto 2011.
- [10] Voip-Info, "Asterisk cmd Vxml", [http://www.voip-
info.org/wiki/view/Asterisk+cmd+Vxml](http://www.voip-info.org/wiki/view/Asterisk+cmd+Vxml) , fecha de consulta: 15 agosto 2011.
- [11] W3C-Office, "Ejemplo: VoiceXML",

<http://www.w3c.es/presentaciones/2004/1118-seminariommi-ja/22.html> ,

fecha de consulta: 15 agosto 2011

[12] Opencorp, "Características y Funcionalidades de Asterisk",

<http://opencorp.co/index.php/productos/37-asterisk/66-caracteristicas-y-funcionalidades-de-asterisk> , fecha de consulta: junio 2011

[13] Voip-Info, "Asterisk cdr mysql", [http://www.voip-](http://www.voip-info.org/wiki/view/Asterisk+cdr+mysql)

[info.org/wiki/view/Asterisk+cdr+mysql](http://www.voip-info.org/wiki/view/Asterisk+cdr+mysql) , fecha de consulta: 20 agosto 2011.

[14] CMUSphinx, "Overview of CMUSphinx toolkit",

<http://cmusphinx.sourceforge.net/wiki/tutorialoverview> , fecha de consulta: 30 julio 2011.

[15] CMUSphinx, "CMUCLMTK Development",

<http://cmusphinx.sourceforge.net/wiki/tutorialoverview> , fecha de consulta: 23 julio 2011.

[16] CMUSphinx, "Building Language Model",

http://cmusphinx.sourceforge.net/wiki/tutoriallm#building_statistical_language_model_using_cmucmtk , fecha de consulta: 2 julio 2011.

[17] ECE Lab, "Natural Language Understanding (NLU)", <http://ecelab.com/natural->

[language-u.htm](#) , fecha de consulta: 30 julio 2011.

[18] All about Linux, "Festival - A text to Speech Systhesis",

<http://linuxhelp.blogspot.com/2006/01/festival-text-to-speech-synthesis.html> ,

fecha de consulta: 9 julio 2011.

[19] , ArticulosInformativos.com, "Qué es Perl",

http://www.articulosinformativos.com/Que_es_Perl-a1106606.html , fecha de

consulta: 8 agosto 2011.

[20] MisRespuestas.com, "Qué es PHP", <http://www.misrespuestas.com/que-es->

[php.html](http://www.misrespuestas.com/que-es-php.html) , fecha de consulta: 24 agosto 2011.

[21] Smarty, "Smarty Tampla Engine" ,

<http://www.smarty.net/docsv2/es/what.is.smarty.tpl> , fecha de consulta: 8 agosto

2011.

[22] Voip Mexico, "Qué es un softphone", <http://www.voipmexico.net/voip-mexico->

[2/que-es-un-softphone](http://www.voipmexico.net/voip-mexico-2/que-es-un-softphone) , fecha de consulta: 4 mayo 2011.

[23] Voip-info.org, "Estimaciones de número de llamadas concurrentes a un servidor

Asterisk",

<http://www.voip-info.org/wiki/view/Asterisk+dimensioning> , fecha de consulta:
20 agosto 2011.

[24] Julian Dunn's Journal, "Sphinx: An Open Source Speech-to-Text Engine",
<http://www.quezada.com/staff/julian/2006/08/29/sphinx-an-open-source-speech-to-text-engine/> , fecha de consulta: 28 julio 2011

[25] Redial: Interactive Telephony, "Speech Recognition",
<http://itp.nyu.edu/~sve204/redial/week9.html> , fecha de publicación: octubre
2006

[26] Sourceforge, "Asterisk PocketSphinx Integration",
<http://sourceforge.net/projects/cmuspinx/forums/forum/5471/topic/4007682>
fecha de consulta: 28 julio 2011

[27] Gouvá Evandro, Chan Arthur, Ravishankar Mosur, Rosenfeld Ronald, Singh Rita, Sun Yitao, Seltzer Mike, "The Hieroglyphs: Building Speech Applications Using CMU Sphinx and Related Resources",
<http://speech.tifr.res.in/tutorials/sphinxDocChan070111.pdf> , fecha de
publicación: 11 marzo 2007, Pág. 8 y 18

- [28] Asterisk, "Re: [Asterisk-Dev] Sphinx Integration", <http://www.mail-archive.com/asterisk-dev@lists.digium.com/msg03067.html> , fecha de publicación: 10 abril 2004
- [29] SyedNetworks, "Asterisk integration with Sphinx Voice recognition system", <http://www.syednetworks.com/asterisk-integration-with-sphinx-voice-recognition-system> , fecha de consulta: 6 agosto 2011
- [30] Speech at CMU, "Sphinx Knowledge Base Tool", <http://www.speech.cs.cmu.edu/tools/lmtool.html> , fecha de publicación: 26 enero 2010
- [31] Asterisk, "Asterisk Downloads", <http://www.asterisk.org/downloads> , fecha de consulta: 3 junio 2011