



FACULTAD DE INGENIERIA Y CIENCIAS AGROPECUARIAS

**Reconocimiento de Comandos de Voz para Control de las Funciones  
Multimedia Dentro del Hogar**

Trabajo de Titulación presentado en conformidad a los requisitos establecidos  
para optar por el título de Ingeniero en Sistemas de Computación e Informática

**Profesor Guía**

Ing. Fernando Solís

**Autor**

José Adrián Salazar Aldaz

**Año**

**2012**

## DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos para un adecuado desarrollo del tema escogido, y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

Fernando Solís

Ingeniero

180300507-1

## DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen derechos de autor vigentes.”

José Adrián Salazar Aldaz

180374516-3

## AGRADECIMIENTO

Agradezco a mis amigos y a mi familia, con especial mención a mis padres, quienes han sabido brindarme todo su apoyo durante esta [larga] carrera. Agradezco también a mis profesores que han logrado cultivar en mí el instinto investigativo y las ganas de siempre ir más allá de lo que está establecido.

## DEDICATORIA

Una persona especial me dijo un consejo muy valioso: "Siempre termina lo que comiences". Este trabajo se lo dedico a esa persona especial, Vanessa Santillán.

## RESUMEN

La tecnología de reconocimiento de voz está disponible desde hace varias décadas, siendo uno de sus principales usos la implementación de *call centers* automatizados, programas de dictado y uso en teléfonos celulares y sistemas de navegación por GPS.

El presente trabajo de investigación describe los conceptos básicos que se manejan en el desarrollo de un sistema de reconocimiento de voz, y presenta una implementación en la cual a través de comandos de voz, el usuario de un hogar promedio podrá controlar las funciones multimedia de su PC y también controlar su televisor LCD o Plasma compatible.

## **ABSTRACT**

Speech Recognition technology has been present during the past few decades, in the form of various implementations like automated call centers, speech dictation programs and embedded systems like smart phones and GPS guided navigation systems.

The present research job describes the basic concepts covered in the development of a speech recognition system, and proposes an implementation on which, the average home user will be able to control through voice commands the multimedia functions from his/her home PC and also will control his/her compatible LCD and Plasma TV Set.

## ÍNDICE

<b>1. CAPÍTULO I</b>	<b>1</b>
1.1. Presentación	1
1.2. Antecedentes	1
1.3. Objetivos	2
1.3.1. Objetivo General	2
1.3.2. Objetivos Específicos	2
1.4. Alcance	2
<b>2. CAPÍTULO II – MARCO TEÓRICO</b>	<b>4</b>
2.1. Tecnologías de Procesamiento de Voz	4
2.1.1. Estado del arte	5
2.1.2. Texto a voz	7
2.1.3. Reconocimiento de Voz	13
2.2. Programación para el procesamiento de voz	16
2.2.1. Hardware	16
2.2.2. Sistemas Operativos	17
2.2.3. Lenguajes de Programación	23
2.2.4. Kits de Desarrollo de Software	26
2.3. Microsoft Speech API	28
2.3.1. Componentes Principales	28
2.3.2. Speech Recognition Engine	29
2.3.3. Text to Speech	31
2.3.4. Integración desde Visual Studio	31
2.4. Tecnologías para control de dispositivos en el hogar	33
2.4.1. Protocolos Abiertos	33
2.4.2. Protocolos Propietarios	34
<b>3. CAPÍTULO III – ANÁLISIS Y DISEÑO</b>	<b>35</b>
3.1. Especificaciones	35
3.1.1. Especificaciones Funcionales	35



3.1.2.	Especificaciones Técnicas	36
3.1.3.	Descripción de Clases	36
<b>3.2.</b>	<b>Diseño</b>	<b>52</b>
3.2.1.	Diseño de Base de Datos	52
3.2.2.	Diseño de la Arquitectura Lógica	60
3.2.3.	Diseño de la Arquitectura Física	61
3.2.4.	Diseño de la Interfaz de Usuario	62
3.2.5.	Diseño de la Gramática del Sistema	66
<b>4.</b>	<b>CAPÍTULO IV – IMPLEMENTACIÓN Y PRUEBAS</b>	<b>71</b>
4.1.	Implementación	71
4.2.	Pruebas	82
4.2.1.	Pruebas Unitarias	82
4.2.2.	Pruebas del Sistema	87
<b>5.</b>	<b>CAPÍTULO V – CONCLUSIONES Y RECOMENDACIONES</b>	<b>90</b>
5.1.	Conclusiones	90
5.2.	Recomendaciones	91
<b>6.</b>	<b>BIBLIOGRAFÍA</b>	<b>93</b>
<b>7.</b>	<b>ANEXOS</b>	<b>96</b>

## LISTADO DE IMÁGENES

IMAGEN 2.1 PASOS PARA LA SÍNTESIS DE VOZ.....	7
IMAGEN 2.2 COMPONENTES DE SPEECH API .....	28
IMAGEN 2.3 EJEMPLO DE GRAMÁTICA DE SAPI.....	30
IMAGEN 2.4 ARQUITECTURA DE <i>SYSTEM.SPEECH</i> .....	32
IMAGEN 3.1 DIAGRAMA DE LA SOLUCIÓN.....	36
IMAGEN 3.2 MODELO DE CLASES DE LA CAPA DE ENTIDADES (PARTE1).....	37
IMAGEN 3.3 MODELO DE CLASES DE LA CAPA DE ENTIDADES (PARTE2).....	38
IMAGEN 3.4 CAPA DE ACCESO A DATOS .....	43
IMAGEN 3.5 INTERFACES DE NEGOCIO .....	45
IMAGEN 3.6 CLASES DE NEGOCIO (PARTE 1) .....	47
IMAGEN 3.7 CLASES DE NEGOCIO (PARTE 2) .....	48
IMAGEN 3.8 MODELO DE BASE DE DATOS .....	53
IMAGEN 3.9 ARQUITECTURA LÓGICA DE LA APLICACIÓN.....	60
IMAGEN 3.10 ARQUITECTURA FÍSICA DE LA APLICACIÓN.....	61
IMAGEN 3.11 PANTALLA DE CARGA DEL SISTEMA.....	62
IMAGEN 3.12 PANTALLA PRINCIPAL DEL SISTEMA.....	63
IMAGEN 3.13 PANTALLA DE CONFIGURACIÓN DE LA BIBLIOTECA MULTIMEDIA.....	64
IMAGEN 3.14 PANTALLA DE CONFIGURACIÓN AVANZADA. ....	65
IMAGEN 3.15 MÁQUINA DE ESTADOS FINITO DE LA APLICACIÓN.....	66
IMAGEN 3.16 FLUJO DE DATOS DE LA APLICACIÓN.....	70
IMAGEN 4.1 COMANDOS DE CONTROL REMOTO DE TELEVISORES LG. ....	74
IMAGEN 4.2 INTERFAZ ITELEVISIONSET .....	76
IMAGEN 4.3 INTERFAZ IMULTIMEDIAPLAYER .....	77
IMAGEN 4.4 INTERFAZ IHOMEAUTOMATOR.....	79
IMAGEN 4.5 PANTALLA DE RESULTADOS DE NUNIT.....	82
IMAGEN 4.6 RESULTADO DE LAS PRUEBAS UNITARIAS DEL SISTEMA.....	83

## LISTADO DE TABLAS

TABLA 2.1 COMPARATIVA ENTRE LENGUAJES DE PROGRAMACIÓN.....	26
TABLA 3.1 COMANDOS DE VOZ DEL SISTEMA EN MODO BASE .....	67
TABLA 3.2 COMANDOS DE VOZ DEL SISTEMA EN MODO MÚSICA.....	68
TABLA 3.3 COMANDOS DE VOZ DEL SISTEMA EN MODO TELEVISIÓN.....	69
TABLA 4.1 PARÁMETROS DE CONEXIÓN PARA TELEVISORES LG.....	71
TABLA 4.2 VALORES ASIGNADOS A CADA ENTRADA DE LOS TELEVISORES LG .....	73
TABLA 4.3 COMANDOS USADOS PARA CONTROL DEL TELEVISOR.....	75

# 1. Capítulo I

## 1.1. Presentación

El presente trabajo de tesis tiene como objetivo el implementar un sistema de reconocimiento de voz que controle las funciones multimedia básicas en un hogar. Estas funciones básicas son: control de la televisión y control de la reproducción de la música.

Para llegar a completar exitosamente esta implementación es necesario revisar los conceptos que intervienen en el proceso de reconocimiento de la voz humana, se parte desde la historia y la evolución que ha tenido esta tecnología hasta llegar al estado del arte, es decir, la tecnología actual disponible.

Se examina a un mayor nivel de detalle las dos tareas que intervienen en un sistema de comandos por voz: la síntesis de voz, y el reconocimiento de voz.

Se realiza también un corto análisis de la tecnología disponible en el mercado y que se podría utilizar en el reconocimiento de voz: hardware, es decir, micrófonos y *arrays* de micrófonos, y software, es decir, sistemas operativos y lenguajes de programación y las facilidades y dificultades que estos ofrecen al momento de diseñar e implementar una aplicación de reconocimiento de voz.

Por último se propone una implementación del sistema propuesto en el título de esta tesis, siguiendo una metodología de desarrollo rápida e incremental, y se documentan las diferentes etapas seguidas en el ciclo de desarrollo del software desarrollado.

Finalmente, se realizan las pruebas respectivas del sistema que arrojan las conclusiones presentadas al final del presente documento.

## 1.2. Antecedentes

La tecnología de reconocimiento de voz ha ido evolucionando desde sus nacimientos en la década de los 60's. Desde sus inicios como una tecnología experimental y con fines académicos, hasta convertirse en parte integral de la tecnología de punta: teléfonos celulares, automóviles de gama alta, paquetes de software para la oficina, sistemas expertos en varios campos de la ciencia usan la tecnología de reconocimiento de voz como la manera de interacción con el usuario.

Por otra parte, los sistemas multimedia: tv, música y video, son una pieza clave en todo hogar moderno, así como también, los sistemas computacionales.

Atrás quedaron los días en que poseer un ordenador era un lujo que muy pocos se podían dar; hoy en día, la PC es el centro de entretenimiento y el nexo entre el hogar y el mundo externo a través de internet.

La tecnología de reconocimiento de voz tiene un enorme potencial de aplicaciones en el hogar, que van desde la ayuda para los usuarios con necesidades especiales o usuarios con limitaciones, activando dispositivos con su voz sin necesidad de accionar interruptores, hasta cambiar el paradigma de cómo se controlan los aparatos electrónicos de uso cotidiano: equipo de música, DVD, televisor, etc., liberando al usuario de la necesidad de usar un control remoto para cada equipo o de la necesidad de estar físicamente en frente del dispositivo, para poder ejecutar los comandos deseados.

## **1.3. Objetivos**

### **1.3.1. Objetivo General**

Desarrollar un sistema de reconocimiento de voz que permita controlar las funciones multimedia de audio y video

### **1.3.2. Objetivos Específicos**

- Diseñar e implementar una gramática de comandos dinámica, que permita al usuario controlar a la aplicación, usando para este fin, Microsoft Speech API.
- Diseñar e implementar una aplicación Windows que a través del Speech API, monitoree el input del micrófono en busca de comandos de voz y ejecute la gramática del sistema.
- Implementar el API de comunicación de los televisores LG.
- Implementar el API de Windows Media, para la reproducción de música.

## **1.4. Alcance**

Se deben poner limitantes al proyecto, ya que el universo de los posibles diálogos que podría generarse en la interacción humano-computador es infinito.

El sistema se deberá centrar únicamente a controlar las funciones multimedia del hogar, a través del computador, usando las interfaces de programación y de hardware estándar y que están disponibles en el mercado.

Las funciones multimedia del hogar a controlar serán:

- Música
  - Reproducción de Música (Play/Pause/Stop/Next/Previous).
  - Búsqueda por género.
  - Búsqueda por artista.
  - Búsqueda por álbum.

- Control de volumen.
- Televisión
  - Encendido/Apagado
  - Selección de fuente (AV/TV/Cable/PC/HDMI)
  - Selección de canal.
  - Control de volumen.

Las interfaces de programación que el sistema utilizará serán:

- Microsoft Speech API, versión 5.1 o superior.
- Windows Media Player SDK, versión 10 o superior.
- LG Television API, para sus series de televisores Plasma y LCD, con soporte de comandos vía RS-232C.

El ambiente para el cual el sistema está diseñado es el siguiente:

- Sistema Operativo: Microsoft Windows XP, Windows Server 2003, Windows Vista Business/Ultimate, Windows Server 2008, Windows 7. Con el último *service pack* instalado.
- Base de datos SQL Server 2008 Express.
- Plataforma de Desarrollo Microsoft .NET Framework 3.5 SP1.
- Microsoft Speech Recognizer Engine v5.1 o superior. (Viene como parte del sistema operativo).
- Windows Media Player v10 o superior.

El sistema no será responsable de la efectividad en el reconocimiento de la voz, ya que ésta se ve influida por varios factores:

- Ruido ambiental.
- Calidad del micrófono usado.
- Potencia de cómputo disponible por parte del procesador.
- Claridad con la que el usuario hable.
- Entrenamiento del perfil de voz del hablante.

Sin embargo, se espera una certeza del reconocimiento de la voz equivalente al 80% de aciertos.

El reconocimiento de voz se la realizará en el lenguaje estándar disponible en todos los computadores, que es: Inglés – Estados Unidos, sin embargo, la retroalimentación de voz desde el computador hacia el usuario (denominado, *Text to Speech*) se la podrá realizar en Inglés – Estados Unidos o en Español.

## 2. Capítulo II – Marco Teórico

### 2.1. Tecnologías de Procesamiento de Voz

La tecnología de procesamiento de voz no es nueva.

No es coincidencia que la llegada del transistor (por consecuencia, los primeros equipos computacionales), y los primeros trabajos de investigación en el campo de reconocimiento de voz, se remonten a la misma década: los 50's.

El reconocimiento de voz es una tarea compleja debido a que la voz es la forma natural de comunicación del ser humano, por tanto, la habilidad humana para entender el lenguaje es innata.

Si consideramos las distintas variaciones de frases que se pueden presentar al momento de procesar la voz humana, y si incluimos factores como: entonación, nivel de la voz, condición buena o mala de salud, fatiga, edad, timbre, sexo; podemos tener una idea de la enorme tarea que se ha venido realizando a lo largo de 60 años de investigación, la cual, ha producido varios algoritmos y modelos de procesamiento de voz.

Una breve revisión de esta investigación podría resumirse:

En los principios de los años 50, los laboratorios BELL desarrollaron un sistema de reconocimiento de voz que podía reconocer dígitos independientes.

De forma paralela, los laboratorios RCA desarrollaron un sistema de reconocimiento para un solo hablante, que era capaz de reconocer hasta 10 sílabas.

El *University College* de Inglaterra desarrolló un reconocedor fonético, capaz de reconocer 30 fonemas (alfabeto) y por otra parte, en el *Lincoln Laboratory* del MIT se desarrollo un reconocedor de vocales, que era independiente de quien hablaba.

Durante los años 60, se crearon algunas herramientas básicas para desarrollo de sistemas de reconocimiento de voz.

El algoritmo de *Dynamic Time Warping* fue desarrollado en los laboratorios de NEC y también en Vyntsyuk de la Unión Soviética.

En la universidad *Carnegie Mellon* de Estados Unidos, se desarrolló un sistema de reconocimiento de habla continua, HAL 9000 con un reducido vocabulario.

Después, en los 70's, varios sistemas fueron desarrollados, como el sistema de reconocimiento de voz automático de IBM, con un largo vocabulario de palabras.

De la mano, se incrementó la inversión para el desarrollo de los sistemas por parte de los gobiernos, como lo son los proyectos DARPA y HARPY de los Estados Unidos de América.

En la década de los 80's, los primeros algoritmos para reconocimiento de habla continua con grandes bases de datos de vocabularios aparecieron.

Y al mismo tiempo, los modelos matemáticos de Markov y las redes neuronales fueron incluidos dentro del desarrollo de sistemas de procesamiento del habla. Estos tipos de sistemas se conocen como los sistemas SPHINX.

Solamente a partir de la década de los 90's se pueden encontrar los primeros sistemas comerciales de reconocimiento de voz, gracias en mayor parte al desarrollo de computadores personales rápidos y de bajo costo, lo que produjo los primeros programas de dictado.

Finalmente, en los últimos años se ha introducido la tecnología en la mayoría de los sistemas operativos, sistemas de telecomunicación, sitios de internet, automóviles de gama alta, aviones civiles y militares, y los usos seguirán aumentando.

### **2.1.1. Estado del arte**

Las últimas décadas han sido de gran beneficio en cuanto a la investigación del reconocimiento de voz.

Los primeros experimentos datan del año 1996, en los cuales, se intentaba reconocer las sílabas pertenecientes a los nombres de los meses del año en inglés.

Se logró obtener un total de 29 sílabas para las cuales, se logró un 84% de precisión del reconocimiento. Un similar trabajo se realizó con las sílabas de los meses en el idioma alemán.

Hauenstein (laboratorios Siemens - Alemania), desarrolló un sistema híbrido, HMM – NN (*Hidden Markov Models*<sup>1</sup> – *Neural Networks*<sup>2</sup>) el cual combinaba redes neuronales con modelos matemáticos de Markov.

---

<sup>1</sup> HMM. Modelo Oculto de Markov, es un modelo estadístico donde existen estados no observados (**ocultos**). Es una variación de un Modelo de Markov en la cual los estados son desconocidos, pero las salidas que estos estados producen si son conocidos. Por lo tanto, la se

El sistema se usó con el propósito de realizar una comparación del rendimiento de dos sistemas de reconocimiento de voz, uno basado en fonemas y otro basado en sílabas.

La conclusión fue que: el sistema que combina fonemas y sílabas es mucho más eficiente que el que usa por separado solo fonemas o solo sílabas.

Más trabajos de investigación realizados en los años siguientes (1997 – 1999) fortalecieron la tesis de que un sistema híbrido puede satisfacer las necesidades de las aplicaciones de reconocimiento de voz.

Esto provocó un gran cambio del paradigma, de pasar de sistemas basados en fonemas, a sistemas basados en reconocimiento de sílabas, lo que demuestra que gran parte del componente fonético del habla humana tiene características silábicas, lo cual volvió necesario el disponer del análisis silábico en los sistemas de reconocimiento de voz.

Webber, en el año 2000, encontró que el *Word Error Rate* – WER (tasa de error de palabras) se incrementa cuando la señal de voz es afectada por el ruido y que dicha tasa de error, se puede reducir si se aumenta el tamaño de la segmentación de la muestra de voz, es decir, nuevamente, si se utiliza análisis de sílabas en lugar de únicamente fonemas.

Para el año 2001, la investigación de Ganapathiraju, Hamaker, Picone, y Doddington dio a conocer que los sistemas de reconocimiento de voz basados en sílabas, logran obtener un performance similar al de los basados en fonemas, ya que, muchos fonemas tienen poca información, lo cual ocasiona que se generen muchas categorías de significados para un mismo sonido, en otras palabras: muchos “falsos positivos”.

Como conclusión, podemos decir que en la actualidad, un sistema de reconocimiento de voz está compuesto por: un motor híbrido, que combina redes neuronales y modelos de Markov, el cual a su vez está en la capacidad de realizar dos tareas fundamentales: reconocimiento de fonemas, y reconocimiento de sílabas.

Dependiendo de la aplicación, se selecciona el algoritmo más adecuado:

---

secuencia de salidas producidas por el modelo dan una idea de la secuencia de estados. El uso de este modelo es común en aplicaciones de reconocimiento de patrones. (Wikipedia)

<sup>2</sup> NN. Redes Neuronales. Son redes de neuronas artificiales, creadas por software que tienen la función de aprendizaje a partir de ejemplos repetitivos. (Wikipedia)



Reconocimiento silábico para las aplicaciones que requieran reconocimiento no continuo del habla (comandos).

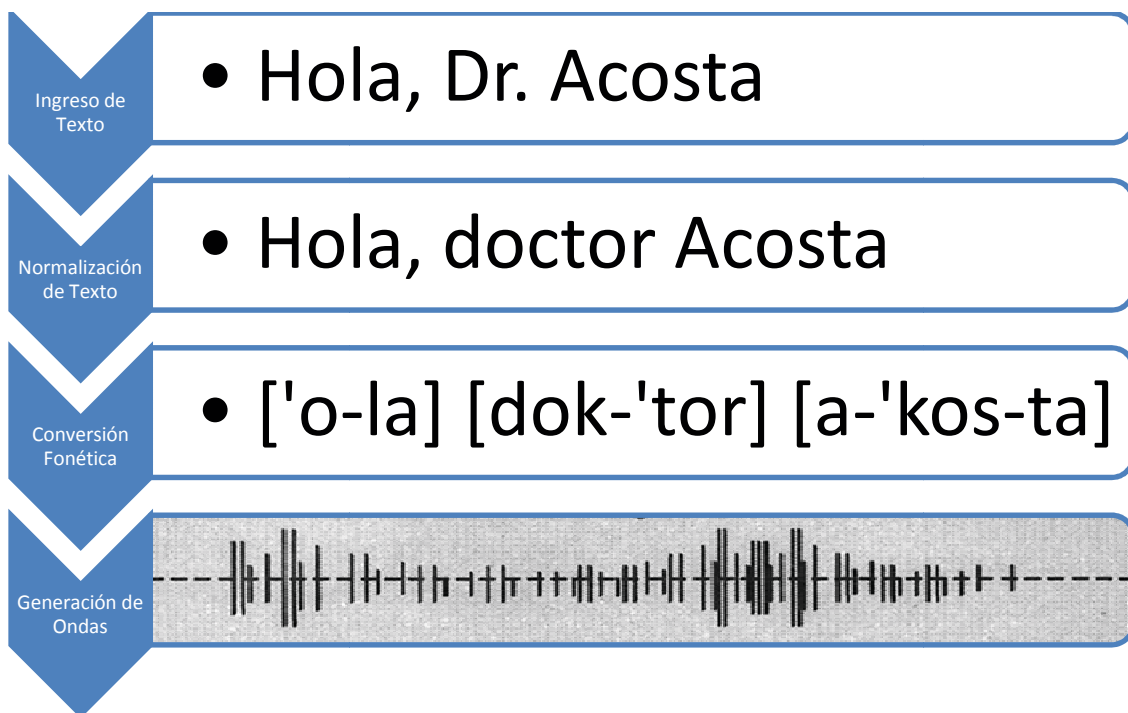
Reconocimiento fonético para las aplicaciones de reconocimiento continuo del habla (dictado).

### 2.1.2. Texto a voz

También conocida como Síntesis del Habla, es la tarea de generar voz artificial, partiendo de un texto.

La Imagen 2.1 nos da una idea de los pasos que sigue un típico sintetizador de voz; pasos que a continuación entraremos a su estudio detallado.

Imagen 2.1 Pasos para la síntesis de voz



**Fuente:** The Voice XML Handbook

#### Ingreso de Texto

El texto a ser hablado es leído desde un buffer o desde un archivo. Una velocidad de entrada de alrededor de 10 bytes por segundo es requerida para dar un promedio de 3 palabras por segundo o 180 palabras por minuto, lo cual es típico para una lectura rápida tal como la practicaría un locutor de radio.

Durante el ingreso de texto, se puede requerir transformación del formato de texto, al formato nativo requerido por el sintetizador. En la mayoría de los

casos, se utiliza Unicode (UTF-16) el cual es capaz de representar la mayoría de lenguajes del planeta.

### **Normalización del texto**

La normalización del texto se encarga de dos tareas principales:

- Expandir abreviaturas.
- Resolver ambigüedades.

Las abreviaturas son la forma más común de ambigüedad. Por ejemplo: “Dr.” podría ser interpretado como “Doctor” (el título académico), “Dronning” (El título de Reina en idioma danés) o “Drive” (referencia de dirección en el idioma inglés).

Los números también requieren de atención especial. “1986” debe ser interpretado como “mil novecientos ochenta y seis” si es que se trata de un año en una fecha, pero por lo contrario, se debe interpretar como “diecinueve - ochenta y seis” si se tratara de una dirección de un domicilio.

El texto “1/2” puede ser interpretado como “mitad” o “un medio” o “primero de febrero” o “dos de enero” dependiendo, de nuevo, del contexto en el cual sea analizado.

Considerando el siguiente texto:

“El Dr. Ayala dio 1/4 de sus ingresos como impuestos durante el 2009, pagando al SRI \$5800.”

Después de la normalización, el texto quedaría así:

“El doctor Ayala dio un cuarto de sus ingresos como impuestos durante el dos mil nueve, pagando al ese-ere-i cinco mil ochocientos dólares.”

Este ejemplo demuestra que se requiere de cierta inteligencia para poder interpretar y normalizar el texto.

### **Búsqueda de excepciones en diccionarios**

Se usan diccionarios para identificar excepciones para reglas comunes de pronunciación.

Esta técnica se hace necesaria cuando, por ejemplo se piensen interpretar nombres propios o palabras que se derivan de otro idioma, las cuales están sujetas a diferentes reglas de pronunciación.

## Conversión a ortografía fonética.

El sintetizador de voz, requiere que se llegue hasta el nivel más bajo de detalle, el fonético, para cada palabra que se desee hablar.

Se pueden usar dos técnicas: basada en reglas (algoritmos) o basada en diccionario (búsqueda).

Cualquier diccionario de bolsillo provee de interpretación fonética. Por ejemplo, al buscar la palabra “gratis” en el diccionario, a continuación de la palabra encontraremos la representación fonética: [gr̩a-t̩s]

Se usan guiones para separar sílabas, la comilla simple indica en que sílaba se encuentra el acento de la palabra; estos son ejemplos de que a través de simbología se puede indicar cómo se debe pronunciar la palabra en cualquier idioma.

El diccionario de lingüística más conocido es el de la IPA (*International Phonetic Alphabet*), el cual posee la más completa colección de fonemas.

En cambio, el modelo basado en algoritmos trata de automáticamente generar esta gramática fonética basada en las reglas propias del lenguaje, por ejemplo, el intérprete intentará convertir cada “c” de la palabra en el sonido fonético de “k”, como en “casa”, a menos que esta letra “c” esté seguida de las letras “e”-“i”; para este caso la “c” se convertirá en una “s”, como por ejemplo en “cerrar”.

Para idiomas con muchas variantes en la pronunciación como son el inglés, francés o alemán, se requiere un extenso número de reglas.

En la práctica, un modelo híbrido es el más adecuado, para el cual, las palabras cuyo modelo fonético de pronunciación no se encuentren por diccionario, se las genera a través de reglas.

## Modificación fonética

El diccionario fonético no siempre va a reflejar la manera en la que una persona pronunciaría un sonido. Por ejemplo, el sonido de la letra “s” en la palabra “mes” no es tan largo como en la palabra “sincero”.

La coarticulación también debe ser considerada. Por ejemplo, una persona no pronunciaría dos veces la letra “o” al leer el número “ciento ochenta”.

Este tipo de problemas son resueltos en este paso. Donde se modifica el tiempo de duración de los fonemas y de las pausas entre fonemas.

## Inflexión

La forma general de la frase, la que determina el tono, la velocidad de habla, y el énfasis de la pronunciación se conoce como inflexión.

Un factor que puede determinar la inflexión de una frase son los símbolos de puntuación de fin de oración, pregunta o exclamación. Considerando los tres ejemplos, la misma frase con diferentes símbolos de puntuación tendrá diferente inflexión. Ejemplo:

Hola, amigo.

¡Hola, amigo!

¿Hola, amigo?

Estas tres frases tienen los mismos componentes fonéticos, pero deben ser pronunciadas de diferente manera.

## Generación de ondas

La etapa final de la síntesis de voz es la generación de ondas de sonidos, partiendo de la representación fonética más la inflexión.

Esto se puede lograr concatenando elementos de sonido (fonemas) pre-grabados, o usando algoritmos acústicos que simulen la acción del tracto vocal, lengua y labios y así, generen ondas de sonido.

Los conjuntos de sonidos pre-grabados habrán de ser cuidadosamente escogidos de una grabación realizada por profesionales.

## Construcción de frases.

Construcción de frases con valores variables.

La principal aplicación de Texto a voz es la construcción de frases informativas. Por ejemplo:

- Call Centers automatizados (bancos y compañías de gran tamaño)
- Buzones de correo de voz
- Servicios Meteorológicos
- Servicios Básicos (luz, agua, teléfono)
- Sitios web compatibles con lectores automáticos para ciegos.

En cada una de estas aplicaciones se tendrá que simular la respuesta humana usando frases pre-definidas, como por ejemplo:

El balance de su cuenta es **ocho** dólares y **cero** centavos.

En el ejemplo anterior, el valor de \$ 8.00 constituye la parte variable de la frase.

### **Componentes de una frase**

Típicamente las frases pueden tener un mensaje genérico, donde uno o más espacios dentro de la frase deben ser llenados por un valor.

Por ejemplo:

“Usted tiene \_\_\_\_ mensajes nuevos.”

En este ejemplo, el espacio en blanco es llenado por un valor numérico. Normalmente se reproduciría la primera parte del mensaje, seguido del valor sintetizando la voz y a continuación se reproduciría la parte final del mensaje.

Si se asume que hay un número finito de posibles valores por llenarse, se podría optar por grabar de corrido cada uno de los mensajes y no tener que concatenar grabaciones. Esto produciría resultados más naturales.

Sin embargo, si se tratan de muchos valores por grabar, la relación coste / beneficio sería alta.

### **Números Enteros**

Para sintetizar una cifra cualquiera en el idioma español, es necesario realizar combinaciones de las siguientes palabras.

- Cero, uno, dos ... nueve, sete, nove
- Diez, once, doce... quince
- y
- Veinte, treinta, cuarenta ... noventa
- Cien, ciento, cientos, quinientos
- Mil
- Millón, millones
- Billón, billones

### **Números Ordinales**

Esta clase de números sirven para expresar orden. Se requiere de gran cantidad de palabras para representar los números ordinales del idioma español.

- 19 primeros ordinales: primero, segundo... décimo noveno.
- Ordinales del 20 en adelante: vigésimo, trigésimo,..., nonagésimo.
- Ordinales del 100 en adelante: centésimo, ducentésimo,..., milésimo.

### **Fechas**

Para el procesamiento de fechas, solo es necesario incorporar 12 palabras más al sintetizador: los meses del año.

Una fecha se lograría uniendo ordinales, cardinales y meses.

### **Horas**

Para procesar horas, es necesario agregar los siguientes elementos al sintetizador:

- De la mañana
- De la tarde
- Medio día
- Media noche
- En punto

### **Dinero**

Para procesar en el contexto de dinero, el sintetizador necesita poseer estas palabras en el diccionario:

- Dólares
- Centavos
- Dólar
- Centavo

El vocabulario va a cambiar dependiendo de la moneda a la que haga referencia el contexto. Euros, Dólares, Nuevos Soles, etc. Lo importante es recalcar que se necesita poder decir el nombre de la moneda y su fracción, tanto en singular como en plural.

### **Cadenas de Dígitos**

Esta parte del análisis está fuertemente ligado a la cultura. Existe variación en la forma de leer cadenas de números entre las diferentes culturas del mundo.

Un ejemplo es la cultura Hispana y la cultura Estadounidense.

Si traducimos al español lo que un norteamericano diría al momento de leer el texto “1 5 9 0 4”, nos encontraremos con una diferencia:

[uno] [cinco] [nueve] [ó] [cuatro]

Se suele reemplazar el número “cero” por la letra “o”.

## Números Telefónicos.

El aspecto a considerar al momento de procesar números telefónicos es que, dependiendo de la convención que se usa en el país donde se encuentre se suelen agrupar los números de distinta manera.

Por ejemplo:

- Ecuador, número local: X – YYY – ZZZ
- Ecuador, número celular: XXX – YYY – ZZZ

Se debe considerar este aspecto antes de realizar la normalización del texto, para de esta manera lograr un efecto más real y menos sintético al oído del usuario.

### 2.1.3. Reconocimiento de Voz

El reconocimiento de voz es una tarea compleja de reconocimiento de patrones en la cual una señal de voz es convertida a señal digital utilizando PCM (*Pulse Code Modulation*<sup>3</sup>) con una frecuencia de muestreo de entre 8 y 16 kHz; la cual es analizada para extraer sus características principales.

Esta primera aproximación es analizada en búsqueda de los límites entre los sonidos fonéticos. Esto le dirá al motor de reconocimiento en donde se encuentran las posibles separaciones entre sílabas. También se obtienen en este paso, los límites entre distintas palabras, es decir, se encontrarán las pausas más largas, desde el punto de vista probabilístico, en la muestra y estas serán tomadas como los límites de cada palabra.

Después, este conjunto de características son analizadas estadísticamente, usando métodos matemáticos como son los modelos ocultos de Markov o *HMM*<sup>4</sup> en los cuales, se obtiene matemáticamente una primera aproximación a la expresión fonética que origina el sonido. Es decir, para cada “ruido”, se generan posibles sílabas que dieron origen al mismo.

Para un mismo “ruido” podrían existir varias combinaciones de letras, conocidas en el lenguaje formal como sílabas, que lo pudieron dar origen.

---

<sup>3</sup> Pulse Code Modulation. Es un método usado para representar digitalmente señales análogas. Es el estándar utilizado para transmitir señales digitales de audio tanto en CD's, DVD's Blu Rays y sistemas de telefonía digital. PCM transmite dos valores básicos, la velocidad de muestreo y la profundidad de muestreo, es decir, cuantos posibles valores de bits contiene cada muestra. (Wikipedia)

<sup>4</sup> HMM: Hidden Markov Models.

Por ejemplo, supongamos que un usuario pronuncia la palabra cabra en el microfono, y se le pasa esto como entrada al software de reconocimiento de voz que esta programado para reconocer palabras en el lenguaje español:

Se van a encontrar cuatro posibles silabas:

Ka – Ca                    tienen una pronunciacion similar.

Bra – Vra    tambien tienen una pronunciacion similar.

Con este conjunto de posibles sílabas, se pasa a comparar estas posibles combinaciones con el modelo de definicion del lenguaje con el que cuenta el motor de reconocimiento. Cada motor esta destinado para un lenguaje en especifico, debido a que cada lengua tiene reglas distintas para pronunciacion, reglas distintas para gramatica, y tambien cada lenguaje tiene su propio conjunto de palabras conocidas, llamada tambien *corpus lingüístico*.

Es por esta razon que el motor de reconocimiento solamente esta habilitado para reconocer palabras en el idioma para el cual fue creado.

Este proceso de encontrar la combinacion de silabas que originó el sonido procesado por el motor de reconocimiento se asemeja en parte al proceso que un software corrector ortografico ejecuta en una palabra.

Se toman las combinaciones de silabas, y se las pasa por el motor de reconocimiento de voz, para comparar con las palabras conocidas del idioma español.

Ka-bra

Ca-bra

Ka-vra

Ca-vra

El motor es capaz de identificar las palabras que no existen en el corpus lingüístico del idioma español, y podrá de esta manera excluir las combinaciones incorrectas, en este ejemplo quedando como unica posibilidad correcta la combinacion: Ca-Bra.

Este es el principio general detrás de un motor de reconocimiento de voz. Existen variantes a este principio dependiendo del tipo de reconocimiento de voz que se piensa ejecutar. Tenemos dos modalidades bien diferenciadas, dos aproximaciones para realizar el reconocimiento de voz:

- Reconocimiento de comandos.



- Reconocimiento de dictado.

## **Modalidades de Reconocimiento de Voz**

### **Modo de comandos**

En el reconocimiento de comandos, el motor debe identificar el ingreso fonético del micrófono contra un listado muy reducido de posibilidades. Es el modo de reconocimiento de voz más generalmente usado, y que más precisión logra.

En este modelo, el corpus lingüístico es delimitado por el desarrollador.

Se logra un alto grado de reconocimiento debido a que el corpus lingüístico es muy pequeño comparado con el universo de palabras que contiene un idioma.

El procesamiento del motor se limita a realizar una comparación entre el corpus y el conjunto de sílabas detectadas en el análisis fonético.

### **Modo de dictado**

Este modo es más complejo ya que no se limita al usuario a un número restringido de frases, ya que como su nombre lo indica, el usuario puede decir al computador, a manera de dictado, cualquier palabra que se encuentre en el corpus del idioma.

El procesamiento que debe realizar el motor de reconocimiento no se limita a la verificación ortográfica sino que también realiza verificación semántica. Aquí entran en juego los modelos de redes neuronales y análisis de probabilidades, que van a decirle al motor, que conjunto de palabras son más susceptibles de aparecer juntas en una frase.

El nivel de precisión en el reconocimiento de voz se reduce, pero se amplía la libertad del usuario.

Una práctica común en los motores de reconocimiento de voz es intentar procesar una entrada fonética como comando primero, y si no se encuentra coincidencia con alguna gramática pre-establecida, se procesa la información en modo de dictado.

## 2.2. Programación para el procesamiento de voz

### 2.2.1. Hardware

#### Micrófonos

Son los dispositivos indispensables en el proceso de adquisición de voz. Ningún sistema de reconocimiento de voz en tiempo real podría funcionar sin la presencia de un micrófono.

Un micrófono funciona a manera de transductor, que convierte las señales acústicas en señales eléctricas.

Este accesorio ha ido evolucionando junto a la ingeniería, logrando obtener cada vez más nivel de fidelidad en el audio que se procesa.

En la actualidad, las tecnologías de micrófono más adecuadas para el reconocimiento de voz son: el micrófono con cancelación de ruido, y el *array* de micrófonos.

#### Micrófono con Cancelación de Ruido

Consta de un micrófono direccional, el cual es posicionado muy cerca a la fuente del sonido del cual se desea grabar el sonido. Por lo general este tipo de hardware se lo emplea en conjunto con los audífonos para formar un *headset*.

#### Array de Micrófonos

Consiste en grabar simultáneamente desde varios micrófonos, los cuales pueden tener diferente grado de sensibilidad ante diferentes frecuencias. El sonido proveniente de estas fuentes es procesado simultáneamente.

Existen numerosas aplicaciones de esta tecnología:

- Extracción de la voz y separación del ruido ambiental.
- Localización de la fuente de sonido en el espacio.
- Grabación de sonido con alta fidelidad.
- Grabación en múltiples canales (sonido *surround*).

El incremento en la calidad de voz al utilizar un *array* de micrófonos es notable, lo que ocasiona que el software de reconocimiento de voz tenga una mejor precisión.

## 2.2.2. Sistemas Operativos

### Microsoft Windows

Windows es el sistema operativo estrella del gigante Microsoft. Disponible en el mercado desde el año 1985. Surgió como reemplazo del hasta ese entonces popular DOS.

Desde sus inicios existieron dos familias de sistemas operativos Windows: Windows 9X y Windows NT.

Windows 9X, es un nombre genérico con el que se conocen a varios sistemas operativos que comparten una misma característica. Todos funcionan a 16 y 32 bits simultáneamente. Todos están basados en MS-DOS por lo que deben mantener el modo de *kernel* (núcleo) de 16 bits para poder ser compatibles con este sistema operativo, pero también ofrecen un modo de 32 bits donde las aplicaciones gráficas pueden ejecutarse.

Los sistemas operativos bajo la categoría de Win9X son:

- Windows 1.0 (1985)
- Windows 2.0 (1987)
- Windows 3.0 (1990)
- Windows 3.11 (1992)
- Windows 95
- Windows 98
- Windows Me

En la actualidad las familias de sistemas operativos Win9X están operativamente fuera de servicio. Microsoft ha dejado de dar soporte y desarrollo a esta plataforma, para seguir desarrollando sistemas operativos de la familia Windows NT.

Las versiones de Windows de la familia NT (NT son las siglas para “*New Technology*”) son versiones puras de 32bits y recientemente de 64bits. Fueron creadas para satisfacer la demanda del usuario empresarial, diseñadas para uso dentro de redes corporativas, tanto para servidores como para estaciones de trabajo.

Al ser un rediseño completo de la arquitectura, este sistema operativo se parece más en características a UNIX que a DOS, lo cual le convierte a Windows NT en un robusto sistema operativo para las empresas.

Los sistemas operativos bajo la categoría de WinNT son:

- Windows NT 3.1 (1993)

- Windows NT 3.5 (1994)
- Windows NT 4.0 (1996)
- Windows 2000 Professional/Server
- Windows XP (2001)
- Windows Server 2003
- Windows Vista (2006)
- Windows 7 (2009)
- Windows 8 (aun no lanzado al público al momento de la escritura de este documento).

Microsoft continúa invirtiendo en el desarrollo y en el posicionamiento de la plataforma NT tanto a nivel de hogar como empresarial, por lo que el sistema operativo a elegirse para realizar desarrollo de software debe ser necesariamente la plataforma NT.

Las ventajas de elegir esta plataforma de desarrollo son:

- El nivel de penetración que la aplicación puede tener en el mercado. En el caso de nuestra aplicación, estamos apuntando a llegar al usuario de casa, y el posicionamiento de Windows en el mercado de hogar sigue siendo dominante en comparación con los demás sistemas operativos de escritorio.
- Un vasto número de material bibliográfico disponible para el desarrollador.
- Un gran número de opciones de lenguajes de programación a escoger, por nombrar algunos de los mas populares: Visual Basic, C, C++, Java, C#, Visual C++, entre otros.

## **Linux**

Si nombre en realidad es el acrónimo “Linux Is Not UniX”, es el resultado de un proyecto para crear un sistema operativo de código abierto, basado en Unix.

Su arquitectura es similar a la UNIX, por lo que comparte las ventajas técnicas, la robustez en su seguridad y el rendimiento.

Sin embargo, este sistema operativo no se lo considero para plataforma de desarrollo de este trabajo de tesis debido a que el nivel de penetración en el mercado doméstico que tiene esta plataforma es bajo en comparación con Windows.

Linux, por el contrario tiene una aceptación en el mercado de servidores de más de 80% lo cual dice bastante sobre cuál es el principal consumidor de este sistema operativo: el mercado empresarial.

## Symbian

Es el sistema operativo más popular usado en *smartphones* durante la última década. Fue desarrollado como una iniciativa, financiada principalmente por Nokia. Entre las demás compañías que financiaron el proyecto se puede nombrar: Ericsson, Samsung, LG y otros menores fabricantes de teléfonos móviles.

Este sistema operativo tiene a su vez, tres variantes, denominados “plataformas”. Las cuales están diseñadas para satisfacer las necesidades de distintos segmentos de mercado.

- Serie 40. Destinada a equipos básicos, del mercado económico.
- Serie 60. Destinada a equipos con funciones multimedia y de smartphone.
- Serie 80. Destinada a equipos con funciones avanzadas de smartphone y de comunicación, incluyendo *tablets* lo que lo acerca más a un sistema operativo de PC.

## Android

Android surgió como la propuesta de Google Inc. de un sistema operativo para smartphones. Está basado en un *kernel* de Linux con la peculiaridad que contiene una versión modificada de una máquina virtual de Java.

Cada aplicación que corre en Android en realidad se ejecuta bajo el contexto de una máquina virtual java con recursos limitados.

Tradicionalmente, las primeras generaciones de teléfonos contaban ya con una versión muy pequeña de una máquina virtual de java, donde se podía ejecutar código muy limitado, llamada la KVM o K- Virtual Machine. Se le dio ese nombre ya que está diseñada para operar en equipos con un espacio de memoria de aproximadamente 128KB. Un espacio muy reducido y que refleja la realidad de la escasez de recursos de hardware a los que está restringido el desarrollo de aplicaciones móviles. Esta plataforma se la conoce comercialmente Java Micro Edition. (Sun Microsystems / Oracle, p. 1)

La diferencia entre Android y Java Micro Edition: para este nuevo sistema operativo, se desarrolló una versión de código abierto de una nueva máquina virtual, llamada *Dalvik*. Además esta nueva máquina virtual tiene una

arquitectura diferente en comparación con la KVM, que le permite al sistema operativo, ejecutar varias instancias de *Dalvik* simultáneamente y de manera eficiente. (Wikipedia)

Android provee un API para reconocimiento de voz, desde la versión 3 de su sistema operativo. Las ventajas de optar por esta plataforma para desarrollo de aplicaciones de voz son:

- Gran aceptación en el mercado. El posicionamiento de *Android* en el mercado de los *smartphones* al cerrar el 2011 fue del 50.9% con tendencia al alza, lo que indica que será el sistema operativo móvil dominante del futuro cercano.
- Gran material disponible en internet, en foros y en la misma comunidad de desarrolladores de Google.
- Gratuidad de acceso a las herramientas de desarrollo.
- Fácil acceso al mercado de aplicaciones de Android Market<sup>5</sup>.
- Relativa facilidad para desarrollar las aplicaciones ya que el lenguaje requerido para crear programas para esta plataforma es java, un lenguaje que lleva ya muchos años en el mercado y es ampliamente conocido por la comunidad.

Si bien esto suena alentador para el desarrollo de aplicaciones de voz, las desventajas que presenta son las siguientes:

- La versión 3 de su sistema operativo está disponible solamente en los dispositivos de alta gama y que han salido al mercado después del año 2010.
- El api no garantiza que un motor de reconocimiento de voz esté presente en el dispositivo. Esta implementación depende también del fabricante del dispositivo quien puede modificar el software base de *Android* de acuerdo a sus necesidades.

## iOS

El sistema operativo *iOS* surge como la propuesta de Apple Inc. para ingresar en el mercado de los *smartphones*. Esta propuesta, incluida en el hardware llamado comercialmente *iPhone* revolucionó el mercado al ser el primer teléfono que ofrecía una pantalla relativamente grande, con capacidad de

---

<sup>5</sup> Android Market: Mercado de aplicaciones para dispositivos Android (*smartphones* y *tablet PCs*) donde el usuario descarga e instala aplicaciones, ya sea de forma gratuita o pagando un valor fijado por el desarrollador de la misma.

*multitouch*<sup>6</sup>, una interfaz de usuario y una experiencia de usuario bastante amigable, notablemente debido a la larga trayectoria de la compañía manufacturando su popular reproductor de música, el *iPod* y sus también populares computadores personales Macintosh.

Siendo Apple propietaria del sistema operativo *MacOS* usado en sus Macintosh, utilizó a éste como base para crear el sistema operativo del *iPhone*. Esto convirtió a *iOS* inmediatamente en la nueva plataforma de desarrollo móvil. Al basarse en *MacOS*, que a su vez está basado en Unix, automáticamente los desarrolladores de esta nueva plataforma fueron beneficiados de las siguientes prestaciones: fácil migración de aplicaciones C/C++, soporte de las librerías más comunes de Unix.

Entre las ventajas de elegir este sistema operativo como plataforma de desarrollo para aplicaciones de voz están:

- Gran número de dispositivos circulando en el mercado. Para finales del 2011, *iOS* es el segundo sistema operativo más disponible en el mercado con un 24% de ventas totales de dispositivos en el mundo.
- Soporte de aplicaciones nativas, esto es, se pueden portar código Mac OS, con su respectivo esfuerzo y ejecutarlo en *iOS*, incluyendo, aplicaciones avanzadas como reconocimiento de voz.

Entre las desventajas están:

- El acceso a las herramientas de desarrollo tiene un costo, el cual es anual. Actualmente (mayo 2012) formar parte del programa de desarrolladores de *iOS* tiene el valor de \$99 USD.
- El proceso de distribución de la aplicación es bastante restringido. Únicamente las aplicaciones aprobadas por Apple Inc., pueden aparecer en la tienda de aplicaciones.
- El nivel de dificultad para desarrollar es moderado ya que hay que adaptarse al un dialecto de lenguaje C llamado *objective-C*.
- No existe variedad de lenguajes a escoger, solamente *objective-C*.

## Windows Phone 7

Microsoft ha tenido una minúscula participación del mercado de sistemas operativos para teléfonos celulares con su sistema operativo Windows Mobile

---

<sup>6</sup> *Multitouch*. Tecnología de pantalla táctil que permite detectar múltiples toques simultáneamente en pantalla, a diferencia de la tradicional tecnología que únicamente detecta un toque a la vez.

6.5 (Wikipedia), por lo que el fabricante de software tomo la decisión de dejar de producir software en la línea de este sistema operativo y comenzar desde cero, diseñando un nuevo sistema operativo con una interfaz nueva y funcionalidades para competir contra los líderes del mercado: iOS y Android.

Este sistema operativo fue liberado al público a finales del año 2010. Está basado en una versión modificada de Windows CE y solo estará disponible únicamente para teléfonos celulares, y no para Tablet PC<sup>7</sup> ni PDA<sup>8</sup> como sus competidores anteriormente mencionados.

Las características principales de este sistema operativo son la interfaz de usuario, táctil con un nuevo paradigma llamado Metro UI. El sistema también integra a Zune, el reproductor multimedia portable de Microsoft, creado para competir con iPod. Otro de los sistemas que se integran con Windows Phone es Xbox Live, la plataforma de juegos online de Microsoft, lo que permite a los usuarios propietarios de una consola Xbox, iniciar sesión en Windows Phone e importar su perfil que será muy útil para utilizar todos los servicios online que ofrece Microsoft.

Las facilidades para el desarrollo de aplicaciones que ofrece Windows Phone son:

- Plataforma de desarrollo Microsoft .NET Framework 4, con una vasta librería de clases.
- Lenguaje de programación C# 4.0 con la misma tecnología disponible en aplicaciones desktop.
- Soporte de *Silverlight*<sup>9</sup> y de *XNA*<sup>10</sup>.

---

<sup>7</sup> Tablet PC. Un tipo de computadora móvil que tiene características similares: pantalla táctil, entrada táctil o por *stylus pen* y un sistema operativo especialmente diseñado para aceptar esta entrada de datos.

<sup>8</sup> PDA. Acrónimo para *Personal Digital Assistant*. Es un tipo de dispositivo portátil, que cabe en la palma de la mano, y sirve como agenda u organizador de la información personal.

<sup>9</sup> Silverlight. Es la plataforma de desarrollo para la Web creada por Microsoft para competir con Flash de Macromedia (actualmente Adobe). Consta de un motor en tiempo de ejecución que visualiza la aplicación en cliente y un motor de ejecución específicamente diseñado para animación. El lenguaje de programación es .NET.

<sup>10</sup> XNA. Acrónimo para "XNA is Not an Acronym". Es una plataforma de desarrollo basada en la tecnología .NET que sirve específicamente para crear contenido multimedia y de juegos. Es la evolución de lo que antes se conocía como Direct X.



- Acceso a los servicios en la nube de Microsoft, como son: Microsoft Tell Me, Cloud Drive, Windows Live ID, Windows Azzure.
- Acceso al mercado de aplicaciones Windows Marketplace.

Las desventajas de desarrollar en la plataforma Windows Phone son:

- No soporta aplicaciones nativas.
- No soporta procesamiento de tareas en segundo plano.
- El reconocimiento de voz se realiza en la nube con Microsoft Tell Me, con lo que es requerido siempre conexión 4G o WiFi. El acceso al servidor de TellMe con una conexión 2G, Edge o 3G no es posible.
- El mercado de aplicaciones no está disponible en todos los países, entre ellos Ecuador.

### **2.2.3. Lenguajes de Programación**

#### **Java**

El lenguaje de programación java fue creado por Sun Microsystems en el año de 1995 con la visión de crear aplicaciones que puedan ser portadas a todas las plataformas.

Es un lenguaje puramente orientado a objetos, altamente portable, debido a que el compilador genera micro código que es interpretado por la maquina virtual de java.

Existen implementaciones de maquina virtual, para diversas plataformas: Windows, Linux, Solaris, Unix, MacOS, dispositivos móviles como celulares y tablet PCs y también dispositivos de hogar como televisores y teléfonos.

Existe un extenso listado de proyectos de código abierto, IDE's de desarrollo y librerías que cubren muchas disciplinas y muchas aplicaciones basadas en este lenguaje de programación.

Las ventajas de usar java para desarrollar aplicaciones de voz en java son:

- Portabilidad. Java está disponible en gran cantidad de plataformas.
- Plataforma abierta.
- Extensa librería de proyectos de código abierto, extenso numero de API's para desarrollar cualquier tipo de proyecto.
- La popularidad del lenguaje hace que exista extensa cantidad de documentación en el internet, como lo son blogs y foros.

La desventaja de elegir java para desarrollar aplicaciones de voz son:

- Se requiere la presencia de la maquina virtual.
- Java no implementa un motor de reconocimiento de forma nativa, por lo que es el sistema operativo el que debe implementar esta funcionalidad.
- Java no está instalado por defecto en los sistemas operativos del hogar. No se asegura que el cliente tenga los componentes necesarios para ejecutar la aplicación.

## **C++**

Este lenguaje de programación es uno de los más antiguos del mercado. Existente desde la década de los 50 ha evolucionado de ser un lenguaje estructurado, conocido como lenguaje C, a llegar a ser un lenguaje potente orientado a objetos, el actual C++.

El nivel de dificultad que representa el desarrollar para este lenguaje es muy alto. El tiempo necesario para desarrollar una aplicación usando este lenguaje es significativamente mayor que el tiempo que tomaría el desarrollar con un lenguaje más moderno como java o c#.

## **C#**

Este fue desarrollado por Microsoft como su contra propuesta para competir con el lenguaje java. Aparece para el público en el año 2002 con la primera versión de Visual Studio .NET.

La visión que sus desarrolladores tuvieron fue la de crear un lenguaje simple y poderoso, con las mejores características similares a las de C++ y java, para nombrar algunas: estricto control de tipos de datos, similar sintaxis de las instrucciones de control y orientación de objetos. También cuenta con mejoras significativas frente a C++ ya que trabaja bajo el .NET Framework, el cual es un ambiente administrado el mismo que automáticamente se encarga de la correcta administración de la seguridad de la aplicación como el manejo de memoria, tareas que en C++ son dejadas a cargo del programador.

Si bien java presenta características de seguridad al también ejecutar los programas bajo un ambiente administrado, la maquina virtual de java, C# tiene ventaja contra esta plataforma de desarrollo al poder interactuar directamente con el API nativo del sistema operativo, en especial con la tecnología COM la cual es la base de todas las API' s del sistema operativo Windows.

Al tener una sintaxis similar a la de C++ y de java, la curva de aprendizaje es mínima, por lo que un programador familiarizado con estos lenguajes puede

rápidamente empezar a producir aplicaciones escritas en C#. Actualmente existe la versión 4.0 de este lenguaje de programación la cual presenta una serie de innovaciones con respecto a sus competidores las cuales le convierten en un lenguaje moderno y poderoso.

Las ventajas de utilizar este lenguaje son:

- Lenguaje rico en instrucciones, con orientación a objetos y estricto control de tipos de dato.
- Control automatizado de memoria y de seguridad de la aplicación.
- Soporte de punteros y de bloques de aplicación que se ejecuten fuera del contexto administrado.
- Soporte de expresiones de programación complejas, expresiones lambda, funciones anónimas, clases anónimas, clases parciales, consulta integrada en el lenguaje "LINQ"<sup>11</sup>.
- Interacción directa con las interfaces nativas del sistema operativo y con DLL's<sup>12</sup>, ATL's<sup>13</sup>, objetos COM+ y librerías desarrolladas en otros lenguajes, como C++ y Visual Basic 6.

### **Comparativa Entre Lenguajes**

En la siguiente tabla podemos observar un resumen comparativo entre los 3 principales lenguajes de programación considerados en este trabajo de tesis para ser el lenguaje en el cual se desarrolle la aplicación.

En esta comparación se puede concluir que C# será el lenguaje elegido para el desarrollo de la aplicación por las ventajas que tiene esta para integrarse de forma nativa con el API de reconocimiento de voz de Windows, frente a las otras dos opciones.

El manejo automático de memoria, el soporte automático de versiones de sistema operativo de 32 y 64 bits y la riqueza de su sintaxis otorgan otra ventaja del lenguaje con respecto a sus rivales.

---

<sup>11</sup> LINQ: Language Integrated Query. Tecnología de búsqueda integrada en el lenguaje disponible en los lenguajes C# y Visual Basic.NET 3.5. Permite al desarrollador ejecutar instrucciones de búsqueda directamente en colecciones de objetos, arrays, colas, listas, documentos XML y otras más fuentes de datos con instrucciones similares al lenguaje SQL. (Microsoft Corporation)

<sup>12</sup> DLL: Dynamic Link Library. Librería compartida, implementada en los sistemas operativos Windows y OS2 que puede contener clases, recursos y datos en cualquier combinación. (Wikipedia)

<sup>13</sup> ATL: Active Template Library. Conjunto de clases basadas en plantillas que simplifican el acceso a objetos COM desarrollados en C++, desde un lenguaje simplificado como son los lenguajes para scripting, por ejemplo VB Script y JavaScript. (Wikipedia)

También se debe considerar que el nivel del conocimiento del lenguaje por parte del desarrollador del presente trabajo de tesis es avanzado y en este aspecto, se convierte en una ventaja a favor de obtener excelentes resultados en el producto de software finalizado.

**Tabla 2.1** Comparativa entre lenguajes de programación.

Característica	Lenguaje de Programación		
	C++	Java	C#
Administración de Memoria	No	Sí	Sí
Control de Seguridad	No	Sí	Sí
Orientación a Objetos	Sí	Sí	Sí
Interacción con código nativo	Sí	Limitada	Sí
Uso de DLL's	Sí	No	Sí
Control de Tipos de Dato	Sí	Sí	Sí
Soporte de Punteros	Sí	No	Sí
API de Reconocimiento de Voz	Speech API	Java Speech API	Speech API / Microsoft.Speech
API de Sintetización de Voz	Speech API	Java Speech API	Speech API / Microsoft.Speech
Forma de invocar al API	Directa	A través de JNI	Directa
Dependencias para el usuario	Ninguna	JRE + Implementación nativa de un motor de reconocimiento.	*Microsoft.NET Framework

Fuente: Autor

## 2.2.4. Kits de Desarrollo de Software

### Java Speech

La plataforma de desarrollo Java no tiene un motor de reconocimiento de voz per se, pero si brinda al programador la facilidad para escribir aplicaciones de voz con un conjunto de clases e interfaces que modelan esta tarea. Este conjunto de clases e interfaces se las conoce como el JSAPI o Java Speech API.

Estas clases fueron concebidas a manera de envoltura, ya que internamente en el entorno administrado de java, estas clases no tienen implementación alguna.

Es tarea del sistema operativo host a través de objetos nativos el proveer de la implementación de estas tareas de reconocimiento de voz. Java conecta a los objetos y a las interfaces de JSAPI con la implementación nativa a través de JNI<sup>14</sup>.

En el caso en el que se desee desarrollar aplicaciones de reconocimiento de voz para la plataforma Windows, la implementación nativa será Microsoft Speech API. Y las llamadas a este se las hará desde java a través de JNI.

## SAPI

Microsoft ha desarrollado el set de herramientas para tareas de reconocimiento de voz desde las primeras versiones del sistema operativo. Este set de herramientas incluyen en la actualidad motores de reconocimiento y de síntesis de voz, en varios idiomas que incluyen pero no se limitan a: inglés, francés, chino, español, alemán, ruso, japonés, para mencionar algunos.

La evolución de estas dos herramientas se puede diferenciar en dos familias de versiones. Las versiones 1.0 hasta la 4.0 y la versión 5.0 y posteriores.

La diferencia entre estas dos familias de versiones radica en que las primeras versiones, hasta la versión 4.0 eran DLL's que podían ser invocadas directamente desde la aplicación, es decir, los programadores tenían acceso directo a los motores de reconocimiento y de síntesis de voz. Esto implica que el desarrollador debe encargarse de tareas de bajo nivel, como control de memoria, control de instancias, control de concurrencia que normalmente debería ser responsabilidad del sistema operativo.

Desde la versión 5.0 de SAPI, se limita el acceso de la aplicación a los motores de reconocimiento y de síntesis de voz, y se deja esta tarea a cargo del *runtime*: sapi.dll el cual expone a través de COM las interfaces que modelan la tarea de reconocimiento y de síntesis de voz y que deben ser invocadas por las aplicaciones, liberando al programador de las tareas de bajo nivel, y haciendo a la tecnología accesible a cualquier lenguaje que se pueda comunicar con COM, entre estos: C++, C#, Visual Basic 6, Visual Basic .NET entre otros.

---

<sup>14</sup> JNI: Java Native Interface. Es un marco de trabajo en el cual Java puede llamar al sistema operativo y ser llamado desde programas y librerías escritas en otros lenguajes como C o C++. (Wikipedia)

Los programas que usaban versiones de SAPI 4.0 o inferiores no pueden usar las versiones de SAPI 5.0 o superior, ya que las arquitecturas son incompatibles. Es necesario un trabajo de re estructuración del código y también algo de re programación.

El historial de versiones disponibles es el siguiente:

- SAPI 1.0 (1995)
- SAPI 3.0 (1997)
- SAPI 4 (1998)
- SAPI 5.0 (2000)
- SAPI 5.1 (2001)
- SAPI 5.2 (2004)
- SAPI 5.3 (Windows Vista)

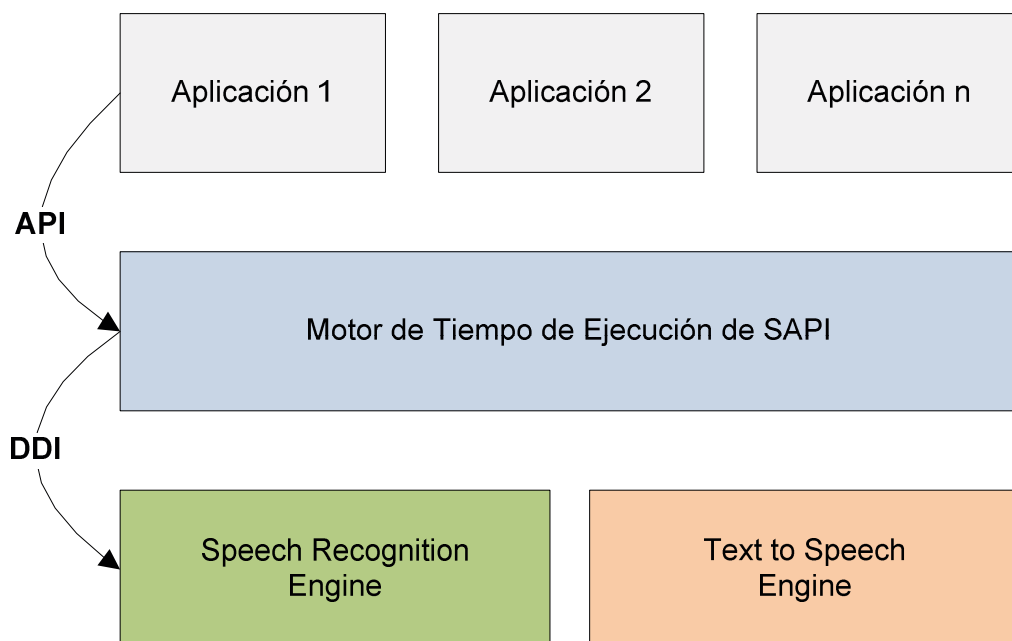
La última versión de es el SAPI 5.4 que viene como parte de Windows 7.

## 2.3. Microsoft Speech API

### 2.3.1. Componentes Principales

*Speech API* se puede resumir en el siguiente diagrama:

**Imagen 2.2** Componentes de Speech API



Fuente: MSDN

Sus componentes principales son:

- Motor de Ejecución de SAPI
- Motor de Reconocimiento de Voz
- Motor de conversión de Texto a Voz

Las aplicaciones Windows hacen llamadas al motor en tiempo de ejecución de SAPI (*runtime*) a través de llamadas API.

El *runtime* a su vez, se encarga de hacer las llamadas de bajo nivel a través de la interfaz DDI (*Device Driver Interface*) a los motores de reconocimiento de voz y de síntesis de voz.

La ventaja de este diseño es que las aplicaciones podrán hacer uso de las funciones de reconocimiento y de síntesis de voz, sin necesidad de conocer la implementación de bajo nivel. Es decir, la misma aplicación podría hacer uso de la versión x del motor de reconocimiento en un computador a, y podría hacer uso de la versión y del motor de reconocimiento en un computador b, sin tener que realizar cambio alguno a su código.

Otra ventaja también es que las operaciones en tiempo real para controlar los motores de reconocimiento y de síntesis de voz, así como el control de eventos y el control del hardware: micrófono y parlantes, ya no están a cargo de la aplicación (y por último, a cargo del programador) sino que es el *runtime* el encargado de esta tediosa tarea.

### **2.3.2. Speech Recognition Engine**

El motor de reconocimiento de voz, soporta dos modalidades: Dictado y comandos.

Este motor además puede funcionar en dos modalidades: Compartido y Dedicado.

En modo compartido SAPI inicializa una sola instancia del motor de reconocimiento, y a través de COM, provee de mecanismos de comunicación de las aplicaciones al motor. Tanto las aplicaciones como el motor de reconocimiento, se ejecutan bajo distintos procesos.

En modo dedicado, cada aplicación puede crear una nueva instancia del motor de reconocimiento de voz. Este uso está destinado para aplicaciones que tengan una mayor demanda de recursos y de control sobre el motor de reconocimiento.

La principal entrada que requiere el motor de reconocimiento es el contexto o también llamado *RecognitionContext*. Cada motor puede tener uno o más

contextos de reconocimiento que se pueden activar y desactivar conforme se los vaya necesitando.

Un contexto está formado por un conjunto de gramáticas o también conocido como *Grammars*, en las cuales el desarrollador modela la forma en la que el usuario puede comunicarse con la aplicación. Este modelado se lo hace a través de reglas, también conocidas como *Rules*, las cuales contienen la definición de más bajo nivel del modelo de conversación que soporta la aplicación.

Una de las reglas más comunes usadas en las aplicaciones de reconocimiento de voz creadas con SAPI, es la regla *OneOf* en la cual, se espera que el motor de reconocimiento reconozca una de entre un listado de posibles palabras.

Las reglas también se pueden anidar, combinar y configurar como obligatorias u opcionales, permitiendo así al desarrollador, crear arboles de reglas dentro de una gramática.

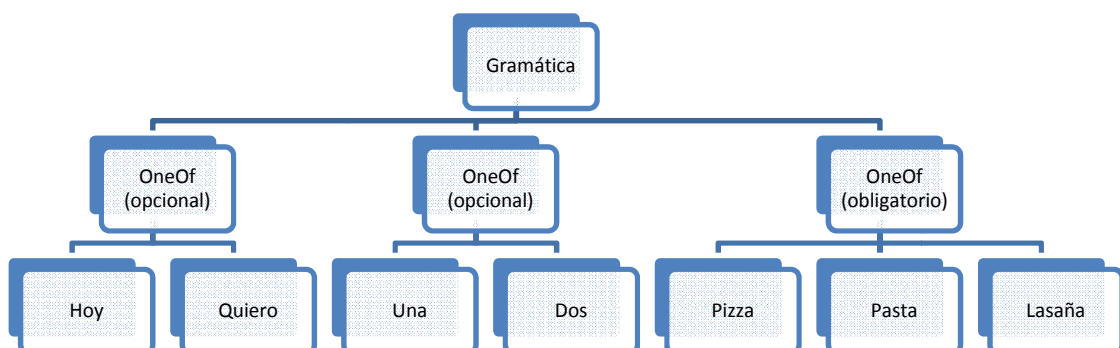
Por ejemplo, en una aplicación de toma de pedidos para una pizzería, el programa le preguntaría al usuario: “¿Cual es su pedido de hoy?”

El usuario puede responder:

- Quiero una pizza
- Hoy quiero pizza
- Quiero pizza
- Pizza
- Dos pizzas

Esto se puede modelar como una combinación de reglas.

**Imagen 2.3** Ejemplo de gramática de SAPI



**Fuente:** Autor



La combinación de reglas, gramáticas y contextos, dan al programador poderosas herramientas para modelar complejas aplicaciones de reconocimiento de voz.

### **2.3.3. Text to Speech**

La síntesis de voz es otro componente principal de SAPI, el cual funciona de forma separada a la de reconocimiento.

Todas las versiones de Windows desde Windows XP tienen un sintetizador de voz instalado. Es posible instalar mas sintetizadores ya que SAPI ha tiene una especificación sobre como implementar componentes compatibles con esta plataforma y hay varias compañías que han desarrollado este tipo de productos bajo el nombre comercial de SAPI *Voices*.

Basta con instalarlas en el equipo y están listas para que cualquier programa que use SAPI, pueda también usar estas voces adicionales.

El proceso que SAPI propone para la síntesis de voz es muy simple.

Se debe instanciar un sintetizador, también conocido como *Synteziser*, e invocar a uno de sus dos métodos principales: *Speak* y *SpeakAsync*.

Lo único que requieren estos métodos es una cadena de texto con la frase que la computadora debe leer.

La diferencia entre estos dos métodos es que el primero bloquea la ejecución del programa hasta que la computadora haya terminado de hablar, y el segundo no realiza este bloqueo.

### **2.3.4. Integración desde Visual Studio**

#### **Visual Studio 2005 / Microsoft .NET Framework 2.0**

La integración del *Microsoft Speech API* desde un proyecto basado en Microsoft .NET Framework 2.0 o inferior, se la realiza a través de una capa llamada COM *Interop*.

Siendo SAPI programado en C++ implementando el modelo de programación COM, es posible importar una referencia COM desde Microsoft Visual Studio.

En ese momento, el IDE de desarrollo, va a examinar las interfaces expuestas por la librería COM del SAPI y va a generar un *assembly* en .NET denominado *Interop*. Este *assembly* expone a .NET las mismas interfaces que el componente COM y se encarga de gestionar las llamadas desde .NET a COM.

## Visual Studio 2008 / Microsoft .NET Framework 3.5

En Microsoft .NET Framework 3.0 en adelante, el reconocimiento de voz está integrado como parte del Framework, ya que se incluye un nuevo espacio de nombres llamado “*System.Speech*”.

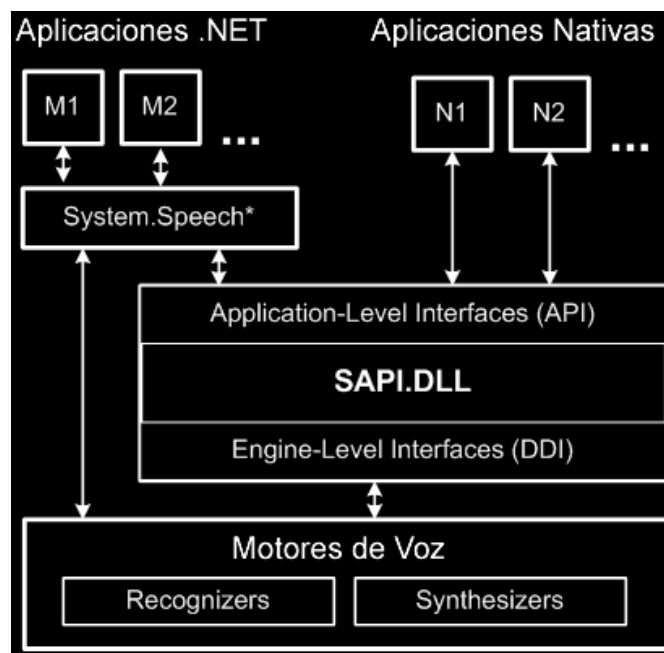
*System.Speech* contiene las clases e interfaces necesarias para programar aplicaciones que hagan uso de las tecnologías de reconocimiento de voz y de conversión de texto a voz.

Sin embargo, hay que notar que la presencia del Framework 3.0 / 3.5 / 4.0 o superior no significa necesariamente que la computadora. La presencia del *namespace System.Speech* significa que las aplicaciones realizadas en tecnología .NET tendrán la habilidad de interactuar con la capa de implementación del sistema operativo sin necesidad de instanciar interfaces COM. Todas las llamadas al API de reconocimiento de voz la hace el *Framework* a través de *managed code*.

### Cambios en la arquitectura:

La siguiente ilustración, muestra las diferentes capas y el modelo de mensajería utilizado entre las misma, dentro de la nueva propuesta denominada *System.Speech*.

Imagen 2.4 Arquitectura de *System.Speech*



Fuente: MSDN

Anteriormente las aplicaciones .NET debían referenciar a SAPI.DLL directamente, lo cual significaba que la aplicación debía pasar por el modelo COM – Interop.

Ahora, las aplicaciones .NET deben hacer uso del espacio de nombres System.Speech que provee de las clases y las interfaces necesarias para implementar el reconocimiento y la síntesis de voz. Esta implementación de System.Speech será la encargada de hacer las llamadas a SAPI o a su vez, podrían interactuar directamente con los motores de reconocimiento y de síntesis de voz.

Los desarrolladores de aplicaciones nativas, desarrolladas en C++ pueden por otro lado, hacer llamadas directas a SAPI, ajustándose al modelo tradicional de programación en COM.

## **2.4. Tecnologías para control de dispositivos en el hogar**

### **2.4.1. Protocolos Abiertos**

#### ***Universal Plug & Play***

Conocido como UPnP es un conjunto de protocolos de red que permiten la comunicación entre dispositivos habilitados para red dentro de un hogar. Por ejemplo: Impresoras, *Gateways*, Computadoras, *Access Points*, Televisores, Equipos de Sonido, etc.

UPNP está destinado al uso dentro del hogar, y no dentro de una red corporativa.

Cada dispositivo de una red UPNP puede descubrir otros dispositivos en la misma red, y también puede anunciarse ante la red.

Cada dispositivo puede exponer servicios ante la red y también puede consumir los servicios que otros dispositivos hayan expuesto.

La ventaja de UPNP es que no está centralizada, en un solo dispositivo, fácil implementación para el usuario de casa, no se requiere configuración. Y prácticamente es transparente para el usuario.

La desventaja es que al ser de fácil implementación, se ha reducido bastante la seguridad, por eso el uso exclusivo en el hogar. Todos los dispositivos están desprotegidos.

Otra desventaja es la dificultad con la que los desarrolladores tienen para interactuar con este set de protocolos. El soporte de los sistemas operativos para UPNP es limitado, no hay soporte, por ejemplo, en Windows XP original, y tampoco en los sistemas empresariales.

La flexibilidad del protocolo también crea dificultad, ya que cada fabricante puede decidir que funcionalidad quiere implementar haciendo difícil la tarea del desarrollador para encontrar una solución compatible con todas las marcas y modelos.

Una desventaja comercial es que el número de dispositivos que soportan UPNP es reducida, únicamente los aparatos domésticos de gama alta los soportan.

### **DLNA**

Es la evolución de UPNP para el uso exclusivo en aparatos multimedia. Se trata de nada adicional al UPNP. Aporta regulaciones sobre la funcionalidad UPNP que deben tener en común los aparatos multimedia que quieran llevar el logotipo DLNA Certified.

### **RS232-C**

Este protocolo es utilizado por varios fabricantes de televisores, con el propósito de que sus aparatos puedan ser controlados a través de un puerto RS232 por un dispositivo externo, muchas veces fabricado por un tercero.

Este protocolo es de simple implementación ya que solo requiere que se comuniquen comandos vía el puerto COM, lo que lo hace un buen punto de inicio para desarrollar aplicaciones de control de aparatos del hogar.

## **2.4.2. Protocolos Proprietarios**

### **SimpLink**

Este protocolo es creado por la empresa LG para comunicar sus aparatos de gama alta, televisores, reproductores de DVD, etc, a través del cable HDMI. El propósito de implementar esta comunicación es lograr que se puedan controlar la mayor cantidad de dispositivos en el hogar con un solo control remoto.

### **Anynet**

Similar al SimpLink, es la propuesta de la empresa SAMSUNG para comunicar los dispositivos multimedia del hogar a través del cable HDMI.

## 3. Capítulo III – Análisis y Diseño

### 3.1. Especificaciones

#### 3.1.1. Especificaciones Funcionales

Se ha de dividir las diferentes especificaciones funcionales en grupos de necesidades, basadas en los objetivos principales detallados en el presente trabajo de titulación, las cuales se pueden resumir en el siguiente listado:

- Se necesita elaborar una librería de clases que permita a la aplicación invocar el API de programación Microsoft Speech API.
- Se necesita diseñar una gramática de comandos que permita al usuario controlar la aplicación.
- Se necesita realizar una aplicación que monitoree el input del Micrófono y responda a los comandos de estos input.
- Se necesita tener control de las funciones básicas de un televisor LCD o Plasma de la marca LG.
- Se necesita controlar el reproductor Windows Media a través de la aplicación.

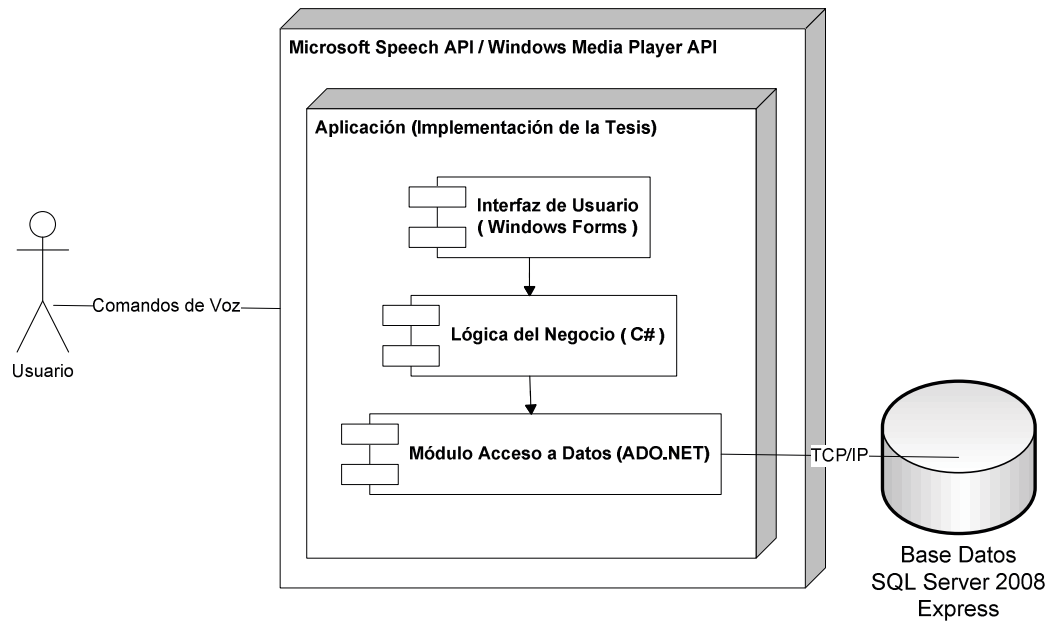
Para cubrir estas necesidades se van a definir los siguientes requerimientos funcionales, siguiendo el estándar IEEE830.

El documento de especificaciones se encuentra en la sección de anexos del presente trabajo de tesis.

### 3.1.2. Especificaciones Técnicas

#### Diagrama de la solución

Imagen 3.1 Diagrama de la solución.



Fuente: Autor

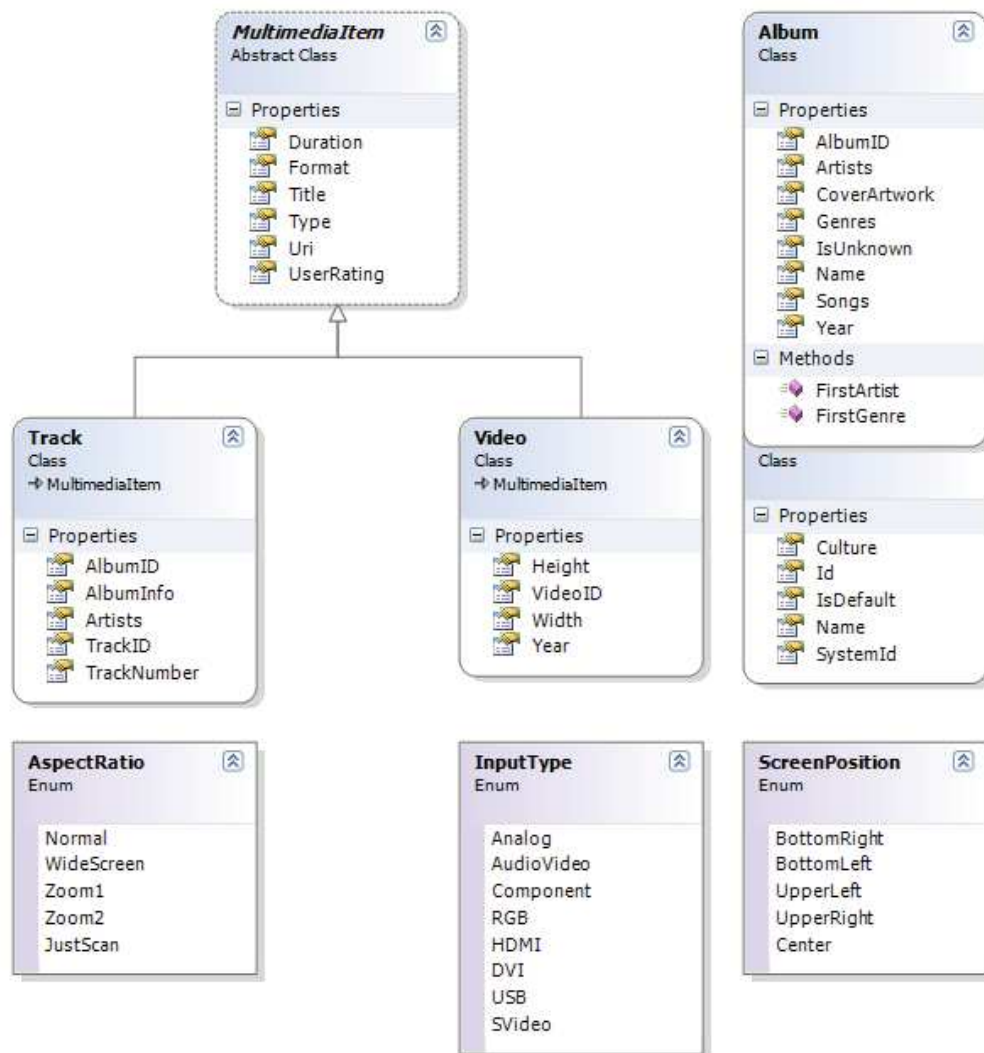
La solución propuesta contará con un diseño en n-capas, una interfaz de usuario, una capa de negocios que será la encargada de orquestar la interacción del programa con la base de datos y las demás interfaces de programación.

### 3.1.3. Descripción de Clases

#### Capa de Entidades

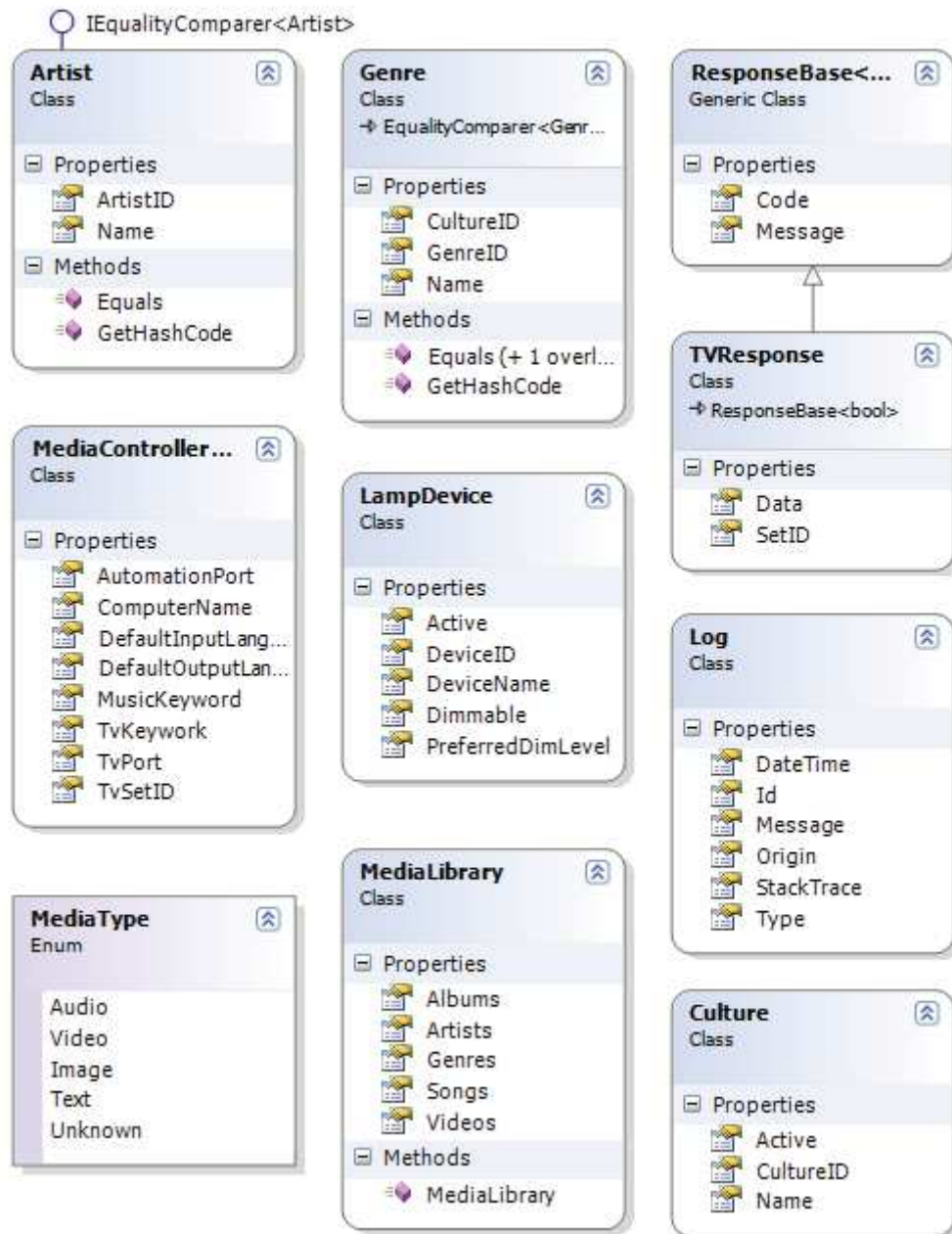
Esta capa de la solución contendrá clases que modelan los datos del sistema de acuerdo al dominio del problema, es decir, no se habla de tipos de datos básicos como son textos y números, sino mas bien se habla aquí de objetos que modelan conceptos propios del problema, como lo son, usuarios, televisores, archivos de música, comandos de voz, etc.

Imagen 3.2 Modelo de clases de la capa de Entidades (parte1)



Fuente: Autor

Imagen 3.3 Modelo de clases de la capa de Entidades (parte2)



Fuente: Autor

A continuación, se describe el propósito de cada clase y también se detallan los atributos de cada clase con su tipo de dato y su significado dentro de la solución.



Clase		MultimediaItem
Descripción		Clase abstracta que encapsula las propiedades básicas de un archivo multimedia, ya sea de audio o video.
Propiedades		
Duration	int	Duración en milisegundos del contenido del archivo.
Format	String	Formato MIME <sup>15</sup> del tipo de archivo.
Title	String	Título del video o de la canción.
Type	String	Tipo de archivo (Audio, Video, Foto, Desconocido)
Uri	String	Dirección URL o dirección física del archivo en disco.
UserRating	bool?	Calificación: positiva ( <i>true</i> )/negativa ( <i>false</i> )/neutral ( <i>null</i> ).

Clase		Album
Descripción		Clase que encapsula las propiedades de un álbum musical.
Propiedades		
AlbumID	Int	ID del álbum en la base de datos.
Artists	Artist[]	Array de artistas que tocan en el álbum.
CoverArtwork	String	URL de la imagen de la portada del álbum.
Genres	Genre[]	Generos musicales asociados al álbum.
IsUnknown	Bool	Indica si el álbum es desconocido.
Name	String	Nombre del álbum.
Songs	Track[]	Array de canciones que contiene el álbum.
Year	Smallint	Año de publicación del álbum.
Métodos		
FirstArtist()	Artist	Obtiene el primer artista asociado al álbum.
FirstGenre()	Genre	Obtiene el primer género asociado al álbum.

Clase		Track : MultimediaItem
Descripción		Extiende la clase MultimediaItem con propiedades que pertenecen a un archivo de música.
Propiedades		
AlbumID	Int	ID del álbum al que pertenece la canción.
AlbumInfo	Album	Información del álbum.
Artists	Artist[]	Array de artistas que intervienen en la canción.
TrackID	Int	ID de la canción en la base de datos.
TrackNumber	Byte	Numero de canción en el álbum.

Clase		Video : MultimediaItem
Descripción		Extiende la clase MultimediaItem con propiedades que definen un archivo de video.
Propiedades		
Height	Int	Alto del video, en pixels.
VideoID	Int	ID del video en la base de datos.
Width	Int	Ancho del video en pixels.
Year	smallint	Año de filmación.

---

<sup>15</sup> MIME. Acrónimo de Multi Purpose Mail Extensions, es un estándar de internet que identifica en una cadena de texto plano el tipo de contenido que representa un archivo. La descripción del tipo de contenido de un archivo multimedia podría ser, por ejemplo: "audio/mpeg4" o "audio/wav", etc. (Wikipedia)

Enumeración		AspectRatio
Descripción		Enumeración que define las distintas opciones de aspecto/forma de imagen que soporta una televisión.
Valores		
Normal	1	Aspecto normal, no se transforma la imagen. Formato 4:3
WideScreen	2	Se fuerza que la imagen tenga formato ancho 16:9
Zoom1	3	Se mantiene el formato pero se incrementa el nivel de zoom.
Zoom2	4	Se mantiene el formato pero se incrementa aun más el zoom.
JustScan	5	El televisor detecta automáticamente el mejor formato para la imagen.

Enumeración		InputType
Descripción		Enumera los posibles tipos de entrada de video de un televisor.
Valores		
Analog	1	Entrada análoga, sintonizador de TV/Cable.
AudioVideo	2	Entrada RCA
Component	3	Entrada de Componente
RGB	4	Entrada de PC
HDMI	5	Entrada HDMI
DVI	6	Entrada DVI (Flat Panel)
USB	7	Entrada USB
SVideo	8	Entrada Súper Video

Enumeración		ScreenPosition
Descripción		Enumera las diferentes posiciones en la que una pantalla de modo <i>picture on picture</i> puede ser colocada en la TV.
Valores		
BottomRight	1	Esquina inferior derecha.
BottomLeft	2	Esquina inferior izquierda.
UpperLeft	3	Esquina superior izquierda.
UpperRight	4	Esquina superior derecha.
Center	5	Centro

Clase		Artist : IEqualityComparer<Artist>
Descripción		Clase que representa las características de un artista (de música). Implementa IEqualityComparer<Artist>.
Propiedades		
ArtistID	Int	ID del artista en la base de datos.
Name	String	Nombre del artista.
Métodos		
Equals(Artist)	Bool	Proviene de la interfaz IEqualityComparer y sirve para comparar si la instancia del objeto es igual a otra instancia de la misma clase.
GetHashCode()	Int	Proviene de la interfaz IEqualityComparer y sirve para obtener un número que representa un identificador único de cada instancia.

Clase	Genre : IEqualityComparer<Genre>	
Descripción	Clase que representa un género musical en el sistema. Implementa IEqualityComparer<Genre>.	
Propiedades		
CultureID	String	ID del idioma en el cual se define el género.
GenreID	Byte	ID interno del género musical en la base de datos.
Name	String	Nombre del género.
Métodos		
Equals(Genre)	Bool	Proviene de la interfaz IEqualityComparer y sirve para comparar si la instancia del objeto es igual a otra instancia de la misma clase.
GetHashCode()	Int	Proviene de la interfaz IEqualityComparer y sirve para obtener un número que representa un identificador único de cada instancia.

Clase	ResponseBase<T>	
Descripción	Clase genérica que representa una respuesta del sistema, donde T es el tipo de dato genérico.	
Propiedades		
Code	T	Código de respuesta.
Message	string	Mensaje de respuesta.

Clase	LampDevice	
Descripción	Clase que representa un dispositivo del hogar tipo lámpara, que soporta control vía comandos desde el computador.	
Propiedades		
Active	Bool	Indica si el dispositivo está habilitado desde la base de datos.
DeviceID	String	ID del dispositivo dentro de la red inteligente del hogar
DeviceName	String	Nombre del dispositivo con el cual el usuario lo conoce dentro del sistema.
Dimmable	Bool	Indica si el nivel de luz se puede controlar.
PreferredDimLevel	Byte	Indica el nivel de luz por defecto que el usuario desea activar en el dispositivo.

Clase	Log	
Descripción	Clase que contiene información de log y de depuración de la aplicación.	
Propiedades		
DateTime	DateTime	Fecha y hora del evento.
Id	Int	ID del evento en base de datos.
Message	String	Mensaje del evento.
Origin	String	Lugar del código donde se origino el evento.
StackTrace	String	Datos de error detallado.
Type	String	Tipo de Evento "E" error, "I" información.

Clase	MediaControllerSettings	
Descripción	Clase que contiene las opciones de configuración del sistema.	
Propiedades		
AutomationPort	String	Puerto por donde se accede a la red inteligente del hogar.
ComputerName	String	Nombre de la computadora, usado en reconocimiento de comandos.
DefaultInputLanguage	String	Código del idioma de entrada.
DefaultOutputLanguage	String	Código del idioma de salida.
MusicKeyword	String	Clave del comando para música.
TvKeyword	String	Clave del comando para TV.
TvPort	String	Puerto por donde se envían los comandos a la TV.
TvSetID	Int	ID de la televisión.

Enumeración	MediaType	
Descripción	Enumera los posibles tipos de archivo multimedia.	
Valores		
Audio	1	Archivo multimedia de audio.
Video	2	Archivo multimedia de video.
Image	3	Archivo multimedia de imagen.
Text	4	Archivo de texto.
Unknown	5	Archivo desconocido.

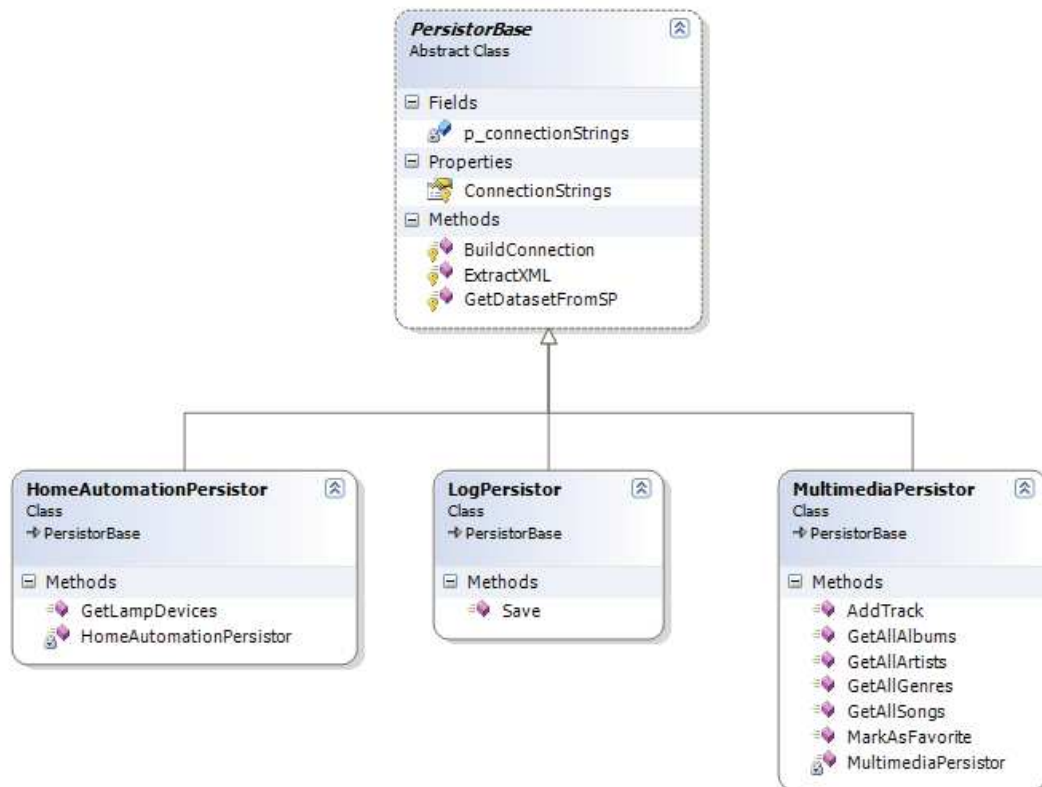
Clase	MediaLibrary	
Descripción	Clase que provee las funciones de biblioteca multimedia del sistema.	
Propiedades		
Albums	Album[]	Array de todos los álbumes del sistema.
Artists	Artist[]	Array de todos los artistas del sistema.
Genres	Genre[]	Array de todos los géneros del sistema.
Songs	Track[]	Array con todas las canciones del sistema.
Videos	Video[]	Array con todos los videos del sistema.
Constructores		
MediaLibrary()	Inicia una nueva instancia de la clase MediaLibrary.	

Clase	Culture	
Descripción	Clase que representa un idioma en el sistema.	
Propiedades		
Active	Bool	Indica si el idioma está habilitado en el sistema.
CultureID	String	ID del idioma en el sistema.
Name	String	Nombre del idioma en el sistema

## Capa de Acceso a Datos

Esta capa contendrá clases que proveerán al sistema con la funcionalidad para leer y escribir en los medios de almacenamiento disponibles, como lo son bases de datos o archivos. En nuestro diseño, solamente se usará como almacenamiento de información el motor de base de datos SQL Server 2008 Express.

**Imagen 3.4** Capa de Acceso a Datos



**Fuente:** Autor

A continuación, se describe el propósito de cada clase y también se detallan los métodos que provee cada clase, con sus parámetros y su tipo de dato de retorno, así como también una breve explicación de su funcionalidad interna.

Clase		PersistorBase
Descripción		Clase abstracta que provee la funcionalidad básica para las clases de acceso a datos.
Propiedades		
ConnectionStrings	StringDictionary	Contiene la colección de cadenas de conexión que maneja el sistema.
Métodos		
BuildConnection(string)	SqlConnection	Construye una conexión a SQL Server, basado en el nombre de la conexión.
ExtractXML(DataSet)	string	Extrae el contenido XML de una DataSet.
GetDataSetFromSP(string)	DataSet	Ejecuta un stored procedure y retorna su resultado como DataSet.

Clase		HomeAutomationPersistor : PersistorBase
Descripción		Extiende la clase PersistorBase para implementar la funcionalidad de almacenamiento en base de datos de las clases relacionadas con el control de la red inteligente del hogar.
Constructores		
HomeAutomationPersistor()	Constructor privado.	
Métodos		
GetLampDevices()	LampDevice[]	Obtiene todos los dispositivos inteligentes del hogar de tipo lámpara, que están registrados en la base de datos del sistema.

Clase		LogPersistor : PersistorBase
Descripción		Extiende la clase PersistorBase y agrega funcionalidad de grabación de logs.
Métodos		
Save(Log)	void	Graba un objeto Log en la base de datos.

Clase		MultimediaPersistor : PersistorBase
Descripción		Extiende la clase PersistorBase y agrega funcionalidad para leer y almacenar los datos de la biblioteca multimedia en la base de datos.
Constructores		
MultimediaPersistor()	Constructor privado.	
Métodos		
AddTrack(Track)	void	Agrega una canción a la base de datos.
GetAllAlbums()	Album[]	Obtiene todos los álbumes de la base de datos.
GetAllArtists()	Artist[]	Obtiene todos los artistas de la base de datos.
GetAllGenres()	Genre[]	Obtiene todos los géneros de la base de datos.
GetAllSongs()	Track[]	Obtiene todas las canciones de la base de datos.
MarkAsFavorite(Track, bool)	void	Marca una canción como favorita.

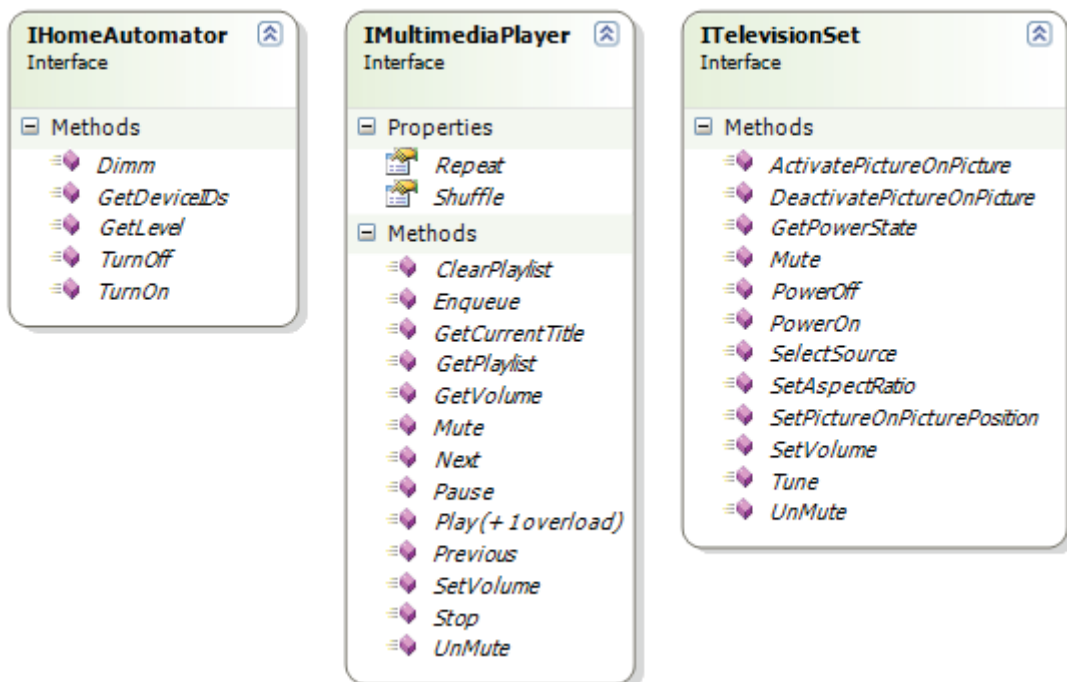
## Capa de Negocio

Esta capa contiene clases que consumen las clases de las capas de entidades y negocio así como también las clases provistas por las diferentes librerías referenciadas por el proyecto, para lograr implementar la completa funcionalidad del sistema.

Se crearon primero interfaces para los 3 elementos externos que se van a controlar desde el sistema, como lo son, un televisor, un reproductor de música y un sistema de control de luces de casa inteligente. Se debe declarar una interfaz ya que en el producto final de software se podría llegar a implementar control sobre varias marcas de tv, se podrían usar varias opciones de reproductor multimedia y se podría optar por usar más de una marca de dispositivos de automatización de hogar inteligente.

La interfaz permite definir las operaciones que el sistema necesita.

Imagen 3.5 Interfaces de Negocio



Fuente: Autor

A continuación, se describe el propósito de cada interface y también se detallan los métodos que declara la interfaz, los cuales deberán ser programados en las clases que implementen la interfaz.

<b>Interfaz</b>	IHomeAutomator	
Descripción	Interfaz que define los métodos que se deben utilizar para controlar las lámparas de una red inteligente de un hogar.	
<b>Métodos</b>		
Dimm(string, byte)	Void	Recibe un ID de dispositivo y un byte que representa el nivel de brillo deseado del dispositivo.
GetDeviceIDs()	String[]	Obtiene los IDs de los dispositivos del hogar.
GetLevel(string)	Byte	Obtiene el nivel de brillo de un dispositivo.
TurnOff(string)	Void	Apaga un dispositivo.
TurnOn(string)	Void	Enciende un dispositivo.

<b>Interfaz</b>	IMultimediaPlayer	
Descripción	Define las operaciones que un reproductor de música debe implementar.	
<b>Propiedades</b>		
Repeat	Bool	Indica si el reproductor de música debe repetir la lista de reproducción.
Shuffle	Bool	Indica si el reproductor de música debe activar el orden aleatorio de reproducción.
<b>Métodos</b>		
ClearPlaylist()	Void	Limpia la lista de reproducción actual.
Enqueue(MultimediaItem)	Void	Agrega un ítem a la lista de reproducción actual.
GetCurrentTitle()	MultimediaItem	Obtiene el ítem que se encuentra en reproducción.
GetPlaylist()	Track[]	Obtiene toda la lista de reproducción.
GetVolume()	Int	Obtiene el nivel de volumen actual.
Mute()	Void	Silencia el volumen del reproductor.
Next()	Void	Toca el siguiente ítem en la lista de reproducción.
Pause()	Void	Pausa la reproducción.
Play()	Void	Inicia/continúa la reproducción.
Play(MultimediaItem)	Void	Toca un ítem en específico.
Previous()	Void	Toca el ítem anterior de la lista de reproducción.
SetVolume(int)	Void	Cambia el volumen del reproductor.
Stop()	Void	Detiene la reproducción.
UnMute()	Void	Quita el silencio y regresa el volumen de la aplicación al estado anterior.

<b>Interfaz</b>	ITelevisionSet	
Descripción	Define las operaciones que debe implementar una clase de control de televisores.	
<b>Métodos</b>		
ActivatePictureOnPicture()	Void	Activa la función de <i>picture on picture</i> .
DeactivatePictureOnPicture()	Void	Desactiva la función de <i>picture on picture</i> .
GetPowerState()	Bool	Consulta el encendido de la TV y devuelve true o false si está encendida o apagada.
Mute()	Void	Pone en silencio la TV.
PowerOff()	Void	Apaga la TV.
PowerOn()	Void	Enciende la TV.
SelectSource(InputType)	Void	Cambia la entrada de video de la TV (HDMI, AV, USB, etc.)
SetAspectRatio(AspectRatio)	Void	Cambia el aspecto de relación de la imagen.
SetPictureOnPicturePosition()	Void	Cambia la posición de la imagen <i>picture on picture</i> .
SetVolume(int)	Void	Cambia el volumen de la TV.
Tune(int)	Void	Cambia un canal.
UnMute()	Void	Quita el silencio de la TV.

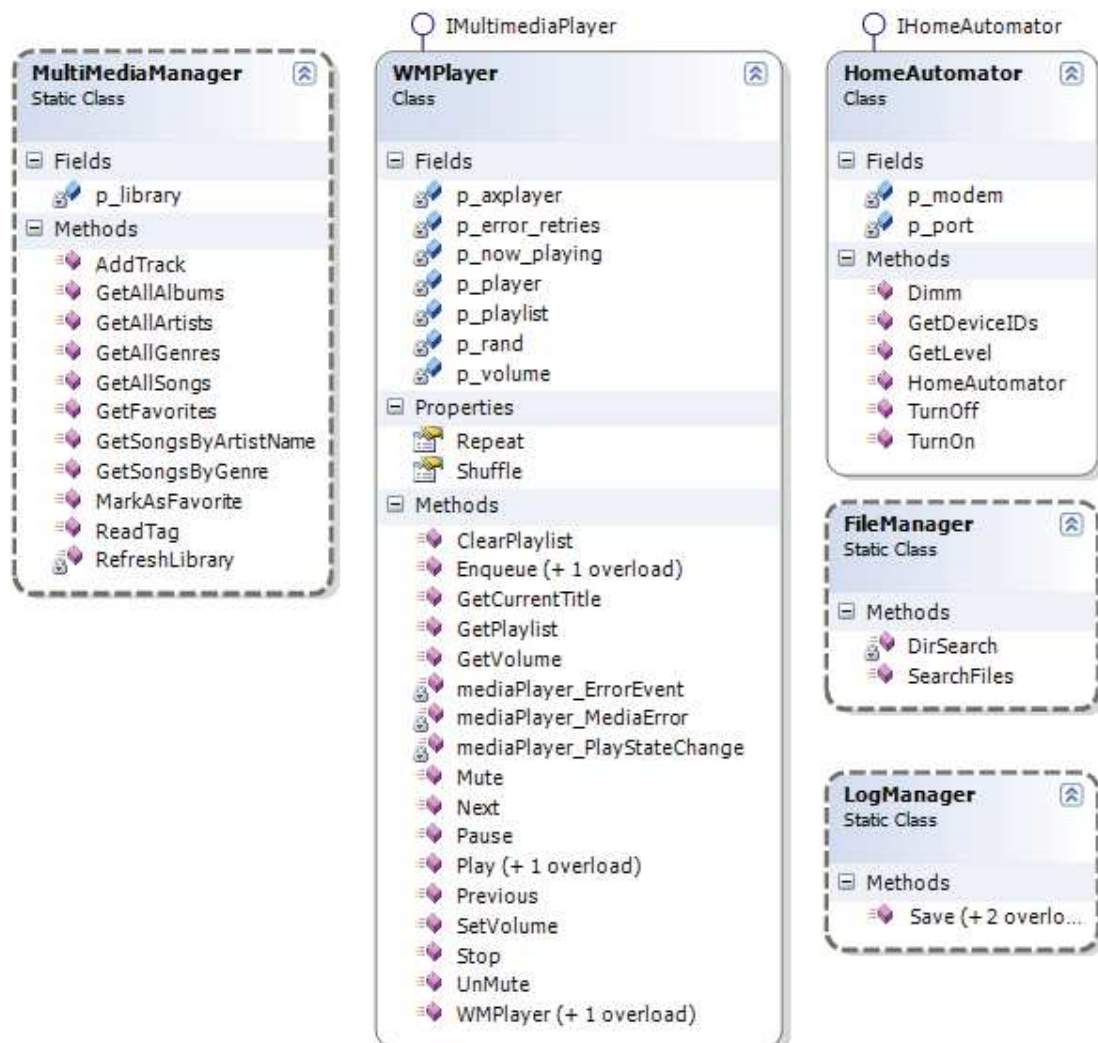


La capa de negocio también contiene clases que implementan estas interfaces, y que realizan también tareas específicas. Como regla general, se crean clases estáticas. Una clase estática por cada grupo de operaciones que se traten del mismo objeto del dominio de la solución, por ejemplo, una clase para operaciones con archivos, una clase para *logging*, una clase para manejo de librería multimedia, etc.

Se prefiere el uso de clases estáticas ya que por lo general las operaciones no van a manejar estado.

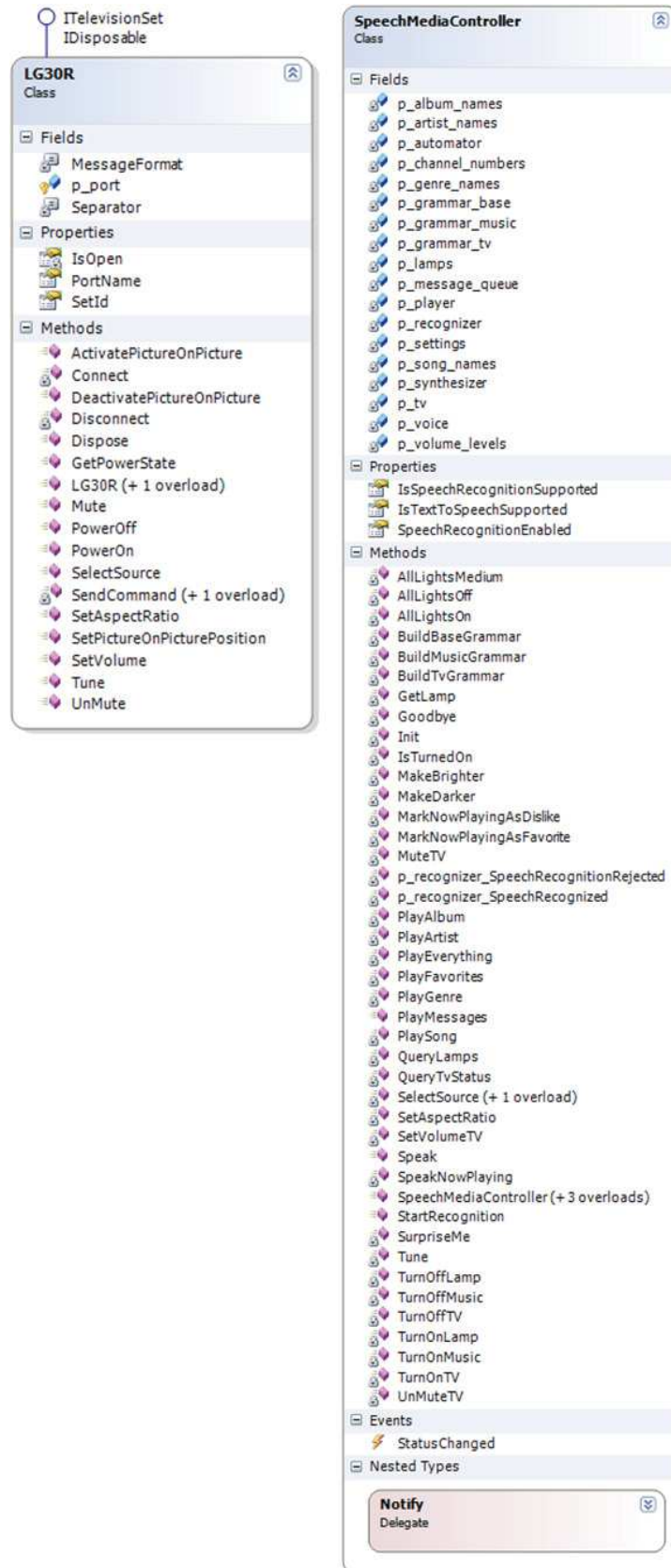
El manejo de estado se dejó a las clases donde este estilo de programación es estrictamente necesario. Por ejemplo, la clase que implementa el control de TV necesita manejar estado, así como la clase que controla la reproducción de música en el sistema, etc.

Imagen 3.6 Clases de Negocio (parte 1)



Fuente: Autor

Imagen 3.7 Clases de Negocio (parte 2)



Fuente: Autor

A continuación, se describe el propósito de cada clase de negocio y también se detallan los métodos que expone, con su tipo de dato de entrada y de salida, además de una breve explicación de la funcionalidad de cada método.

<b>Clase</b>	MultiMediamanager	
Descripción	Provee de funciones de búsqueda sobre la biblioteca multimedia del sistema.	
<b>Propiedades</b>		
p_library	MultiMedia Library	Instancia de la biblioteca multimedia del sistema
<b>Métodos</b>		
AddTrack(Track)	Void	Agrega una canción a la biblioteca multimedia del sistema y la graba en base de datos.
GetAllAlbums()	Album[]	Obtiene todos los álbumes de la base de datos.
GetAllArtists()	Artist[]	Obtiene todos los artistas de la base de datos.
GetAllGenres()	Genre[]	Obtiene todos los géneros de la base de datos.
GetAllSongs()	Track[]	Obtiene todas las canciones de la base de datos.
GetFavorites()	Track[]	Obtiene las canciones marcadas como favoritas.
MarkAsFavorite (Track, bool)	void	Marca una canción como favorita.
GetSongsByArtist Name()	Track[]	Obtiene todas las canciones de un artista.
GetSongsByGenre()	Track[]	Obtiene todas las canciones de un género.
ReadTag()	Void	Lee el <i>ID3 Tag</i> de un archivo multimedia.
RefreshLibrary()	Void	Refresca la librería multimedia con los últimos datos de base de datos.

<b>Clase</b>	HomeAutomator : IHomeAutomator	
Descripción	Implementa la interfaz IHomeAutomator usando la librería de control de dispositivos Insteon y X10 llamada <i>FluentDwelling</i>	
<b>Métodos</b>		
Dimm(string, byte)	Void	Recibe un ID de dispositivo y un byte que representa el nivel de brillo deseado del dispositivo.
GetDeviceIDs()	String[]	Obtiene los IDs de los dispositivos del hogar.
GetLevel(string)	Byte	Obtiene el nivel de brillo de un dispositivo.
TurnOff(string)	Void	Apaga un dispositivo.
TurnOn(string)	Void	Enciende un dispositivo.

<b>Clase</b>	FileManager	
Descripción	Clase que provee de operaciones con el sistema de archivos del computador.	
<b>Métodos</b>		
DirSearch(string, string)	File[]	Busca archivos dentro de un directorio.
SearchFiles()	File[]	Busca archivos dentro de un directorio – recursivo.

<b>Clase</b>	WMPayer : IMultimediaPlayer	
Descripción	Implementa la interfaz IMultimediaPlayer haciendo uso del API de Windows Media Player	
<b>Propiedades</b>		
Repeat	Bool	Activa/Desactiva la repetición de la lista de canciones.
Shuffle	Bool	Activa/Desactiva el orden aleatorio de la lista de reproducción.
<b>Constructores</b>		
WMPayer()	Inicia una nueva instancia del reproductor multimedia con las configuraciones por defecto.	
WMPayer(AxWMPayer)	Inicia una nueva instancia del reproductor multimedia con el reproductor de Windows Media especificado por la capa superior.	
<b>Métodos</b>		
ClearPlaylist()	Void	Limpia la lista de reproducción actual.
Enqueue(MultimediaItem)	Void	Agrega un ítem a la lista de reproducción actual.
GetCurrentTitle()	MultimediaItem	Obtiene el ítem que se encuentra en reproducción.
GetPlaylist()	Track[]	Obtiene toda la lista de reproducción.
GetVolume()	Int	Obtiene el nivel de volumen actual.
Mute()	Void	Silencia el volumen del reproductor.
Next()	Void	Toca el siguiente ítem en la lista de reproducción.
Pause()	Void	Pausa la reproducción.
Play()	Void	Inicia/continúa la reproducción.
Play(MultimediaItem)	Void	Toca un ítem en específico.
Previous()	Void	Toca el ítem anterior de la lista de reproducción.
SetVolume(int)	Void	Cambia el volumen del reproductor.
Stop()	Void	Detiene la reproducción.
UnMute()	Void	Quita el silencio y regresa el volumen de la aplicación al estado anterior.

<b>Clase</b>	LogManager	
Descripción	Clase que provee de funcionalidad de grabación de Logs.	
<b>Métodos</b>		
Save(Log)	Void	Graba un objeto Log en la base de datos.
Save(Exception)	Void	Genera un objeto Log a partir de una Exception y graba esta información en base de datos.
Save(string, string)	Void	Genera un objeto Log a partir de un tipo de evento y un mensaje y graba esta información en base de datos.

<b>Clase</b>	SpeechMediaController	
Descripción	Clase que implementa la funcionalidad completa del sistema, desde esta clase se instancian las demás clases que permiten el control de TV/Música/Luces y se implementan las funciones de reconocimiento de voz las cuales al activarse, enviaran comandos a las clases encargadas del control de cada función del sistema.	
<b>Propiedades</b>		
IsSpeech RecognitionSupported	Bool	Indica si el reconocimiento de voz está soportado en el computador.
IsTextToSpeechSuported	Bool	Indica si la función de conversión de texto a voz está soportada en el computador.
SpeechRecognitionEnabled	Bool	Indica si el reconocimiento de voz está habilitado desde la interfaz de usuario.
<b>Constructores</b>		
SpeechMediaController(Settings)	Inicia una nueva instancia del objeto recibiendo los parámetros de configuración.	
SpeechMediaController(Settings, AxWmPlayer)	Inicia una nueva instancia del objeto recibiendo los parámetros de configuración y una instancia de un reproductor de Windows Media Player.	
SpeechMediaController(ITvSet, IMultimediaPlayer, IHomeAutomator)	Inicia una nueva instancia del objeto recibiendo un puntero a cada una de las interfaces de control de los dispositivos del sistema.	
<b>Métodos</b>		
PlayMessages()	Void	Reproduce los mensajes de depuración de la aplicación.
Speak(string)	Void	Reproduce un texto con la función de síntesis de voz.
StartRecognition()	Void	Inicia el motor de reconocimiento de voz.

<b>Clase</b>	LG30R : ITelevisionSet	
<b>Descripción</b>	Clase que implementa ITelevisionSet para la familia de televisores LG30R.	
<b>Propiedades</b>		
IsOpen	Bool	Indica si el objeto está conectado al TV.
PortName	String	Indica el nombre del puerto
SetId	Int	
<b>Constructores</b>		
LG30R(string)		Inicia una nueva instancia de la clase, especificando un nombre de puerto y usando el SetID por defecto (0).
LG30R(string, int)		Inicia una nueva instancia de la clase, especificando un nombre de puerto y usando el SetID especificado por la capa superior.
<b>Métodos</b>		
ActivatePictureOnPicture()	Void	Activa la función de <i>picture on picture</i> .
DeactivatePictureOnPicture()	Void	Desactiva la función de <i>picture on picture</i> .
GetPowerState()	Bool	Consulta el encendido de la TV y devuelve true o false si está encendida o apagada.
Mute()	Void	Pone en silencio la TV.
PowerOff()	Void	Apaga la TV.
PowerOn()	Void	Enciende la TV.
SelectSource(InputType)	Void	Cambia la entrada de video de la TV (HDMI, AV, USB, etc.)
SetAspectRatio(AspectRatio)	Void	Cambia el aspecto de relación de la imagen.
SetPictureOnPicturePosition( )	Void	Cambia la posición de la imagen <i>picture on picture</i> .
SetVolume(int)	Void	Cambia el volumen de la TV.
Tune(int)	Void	Cambia un canal.
UnMute()	Void	Quita el silencio de la TV.
Connect()	Void	Conecta el objeto con el puerto COM de la TV.
Disconnect()	Void	Desconecta el objeto del puerto COM de la TV.
SendCommand(string)	String	Envía un comando al puerto COM de la TV y retorna la respuesta de la TV.

## 3.2. Diseño

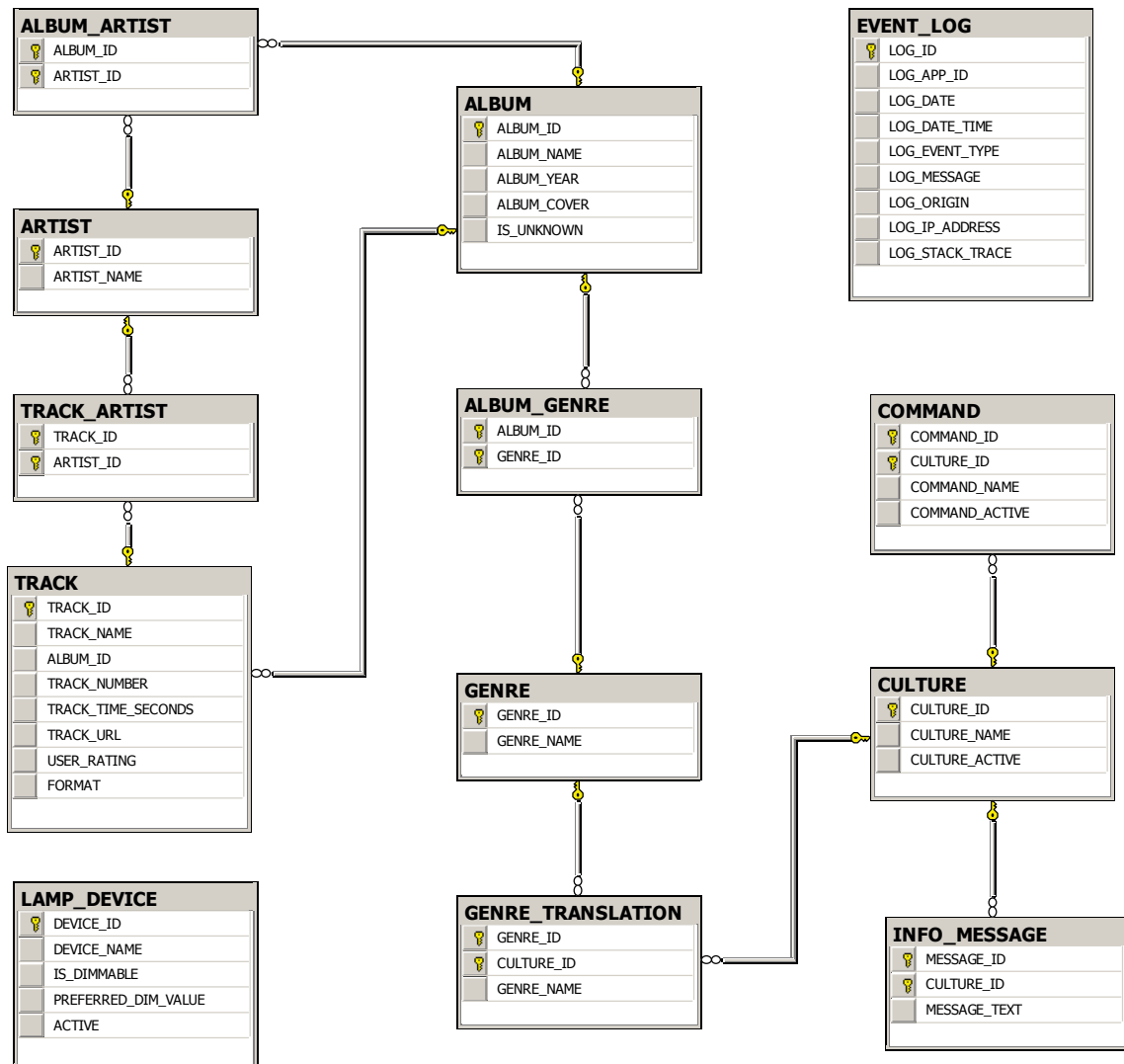
### 3.2.1. Diseño de Base de Datos

#### Modelo de Base de Datos

El sistema contará con una base de datos relacional, la cual se encargara del manejo de la información de la biblioteca multimedia del sistema, llevará un log de los eventos que ocurran en el sistema, y también contendrá el listado de comandos del sistema con su respectiva traducción a los idiomas que soporta el sistema.

A continuación se muestra el modelo lógico de la base de datos.

Imagen 3.8 Modelo de base de datos



Fuente: Autor

## Diccionario de Datos

A continuación se lista cada tabla que compone el modelo lógico y se la describe. Se lista cada columna, con su nombre físico, el tipo de dato que se almacena en la columna, y se encuentran resaltadas con negrita los nombres de las columnas que conforman la clave primaria de la tabla.

Tabla	CULTURE	
Descripción	Contiene los diferentes idiomas soportados por el sistema.	
Columnas		
<b>CULTURE_ID</b>	varchar(5)	ID del idioma dentro del sistema.
CULTURE_NAME	varchar(20)	Nombre del idioma para el sistema.
CULTURE_ACTIVE	bit	Indica si el idioma se encuentra activo para el usuario dentro del sistema.

Tabla	EVENT_LOG	
Descripción	Almacena los logs del sistema, ya sea información de depuración o información de errores.	
Columnas		
<b>LOG_ID</b>	int	ID del log dentro de la base de datos.
LOG_APP_ID	varchar(20)	ID de la aplicación que genera el mensaje.
LOG_DATE	date	Fecha del evento.
LOG_DATE_TIME	datetime	Fecha y hora del evento.
LOG_EVENT_TYPE	char(1)	Tipo de evento.
LOG_MESSAGE	varchar(255)	Mensaje del evento.
LOG_ORIGIN	varchar (100)	Origen del evento.
LOG_IP_ADDRESS	varchar (16)	Dirección IP asociada al evento.
LOG_STACK_TRACE	varchar(MAX)	Información detallada del evento.

Tabla	COMMAND	
Descripción	Contiene el listado de comandos soportados por el sistema con su traducción en varios idiomas.	
Columnas		
<b>COMMAND_ID</b>	varchar(20)	ID del comando dentro del sistema.
<b>CULTURE_ID</b>	varchar(5)	ID del idioma en que está escrito el comando.
COMMAND_NAME	varchar(120)	Texto del comando.
COMMAND_ACTIVE	bit	Indica si el comando esta activo.

Tabla	INFO_MESSAGE	
Descripción	Contiene la traducción de las cadenas de texto que se utilizan dentro de la aplicación.	
Columnas		
<b>MESSAGE_ID</b>	varchar(20)	ID de la cadena de texto.
<b>CULTURE_ID</b>	varchar(5)	ID del idioma de la cadena de texto.
MESSAGE_TEXT	varchar(512)	Cadena de texto.

Tabla	ARTIST	
Descripción	Contiene el listado de los artistas de la biblioteca multimedia del sistema.	
Columnas		
<b>ARTIST_ID</b>	int	ID del artista.
ARTIST_NAME	int	Nombre del artista.



<b>Tabla</b>	ALBUM	
Descripción	Contiene el listado de los álbumes contenidos en la biblioteca multimedia del sistema.	
Columnas		
<b>ALBUM_ID</b>	int	ID del álbum en el sistema.
<b>ALBUM_NAME</b>	varchar(100)	Nombre del álbum.
<b>ALBUM_YEAR</b>	smallint	Año del álbum.
<b>ALBUM_COVER</b>	varchar(100)	URL de la cubierta del álbum.
<b>IS_UNKNOWN</b>	bit	Indica si el álbum es desconocido.

<b>Tabla</b>	ALBUM_ARTIST	
Descripción	Asocia álbumes con artistas.	
Columnas		
<b>ALBUM_ID</b>	int	ID del álbum.
<b>ARTIST_ID</b>	int	ID del artista.

<b>Tabla</b>	ALBUM_GENRE	
Descripción	Asocia álbumes con géneros.	
Columnas		
<b>ALBUM_ID</b>	int	ID del álbum.
<b>GENRE_ID</b>	int	ID del género.

<b>Tabla</b>	TRACK_ARTIST	
Descripción	Asocia artistas con canciones.	
Columnas		
<b>TRACK_ID</b>	int	ID de la canción.
<b>ARTIST_ID</b>	int	ID del artista.

<b>Tabla</b>	GENRE	
Descripción	Contiene el listado de los géneros contenidos en la biblioteca multimedia del sistema.	
Columnas		
<b>GENRE_ID</b>	int	ID del género dentro del sistema.
<b>GENRE_NAME</b>	varchar(100)	Nombre del género.

<b>Tabla</b>	GENRE_TRANSLATION	
Descripción	Contiene las traducciones de los nombres de los géneros en el sistema.	
Columnas		
<b>GENRE_ID</b>	int	ID del género dentro del sistema.
<b>CULTURE_ID</b>	varchar(5)	ID del idioma dentro del sistema.
<b>GENRE_NAME</b>	varchar(100)	Traducción del género en ese idioma.

<b>Tabla</b>	LAMP_DEVICE	
Descripción	Contiene el listado de los dispositivos inteligentes del hogar.	
Columnas		
<b>DEVICE_ID</b>	varchar(10)	ID del dispositivo.
DEVICE_NAME	varchar(100)	Nombre del dispositivo.
ID_DIMMABLE	bit	Indica si el brillo del dispositivo es configurable.
PREFERRED_DIM_VALUE	tinyint	Indica el valor de brillo preferido del usuario.
ACTIVE	bit	Indica si el dispositivo puede ser usado.

<b>Tabla</b>	TRACK	
Descripción	Contiene la información de las canciones de la biblioteca multimedia del sistema.	
Columnas		
<b>TRACK_ID</b>	int	ID de la canción dentro del sistema.
TRACK_NAME	varchar(100)	Nombre de la canción.
ALBUM_ID	int	ID del álbum al que pertenece la canción.
TRACK_NUMBER	tinyint	Numero de pista dentro del álbum.
TRACK_TIME_SECONDS	smallint	Tiempo de la canción en segundos.
TRACK_URL	varchar(255)	URL de la canción.
USER_RATING	bit	Rating del usuario.
FORMAT	varchar(20)	Formato de la canción expresado en MIME.

## Procedimientos Almacenados

<b>Procedure</b>	USP_ADD_ALBUM	
Descripción	Agrega un álbum a la biblioteca multimedia del sistema.	
Retorno	XML	
Parámetros		
@ALBUM_NAME	varchar(100)	Nombre del álbum.
@ALBUM_YEAR	smallint	Año del álbum.
@ALBUM_COVER	varchar(255)	URL al archivo de imagen de la portada del álbum.

<b>Procedure</b>	USP_ADD_ALBUM_ARTIST	
Descripción	Asocia un artista con un álbum.	
Retorno	XML	
Parámetros		
@ALBUM_ID	int	ID del álbum en base de datos.
@ARTIST_ID	int	ID del artista en base de datos.

<b>Procedure</b>	USP_ADD_ALBUM_GENRE	
Descripción	Asocia un álbum con un género.	
Retorno	XML	
Parámetros		
@ALBUM_ID	int	ID del álbum en base de datos.
@GENRE_ID	int	ID del género en base de datos.

<b>Procedure</b>	USP_ADD_ARTIST	
Descripción	Agrega un artista a la base de datos multimedia.	
Retorno	XML	
Parámetros		
@ARTIST_NAME	Varchar(100)	Nombre del artista en base de datos.

<b>Procedure</b>	USP_ADD_FILE	
Descripción	Agrega un archivo multimedia a la base de datos.	
Retorno	XML	
Parámetros		
@FILENAME	varchar(255)	URL del archivo a agregar.

<b>Procedure</b>	USP_ADD_GENRE	
Descripción	Agrega un género a la biblioteca multimedia del sistema.	
Retorno	XML	
Parámetros		
@GENRE_NAME	varchar(100)	Nombre del género a agregarse.

<b>Procedure</b>	USP_ADD_TRACK	
Descripción	Agrega una canción a la biblioteca multimedia del sistema.	
Retorno	XML	
Parámetros		
@TRACK_NAME	varchar(100)	Nombre de la canción.
@ALBUM_ID	int	ID del álbum al que pertenece la canción.
@TRACK_NUMBER	tinyint	Numero de la canción dentro del álbum.
@TRACK_TIME_SECONDS	smallint	Tiempo de duración de la canción.
@TRACK_URL	varchar(255)	URL al archivo de la canción.
@USER_RATING	bit	Rating del usuario.
@FORMAT	varchar(20)	Formato MIME de la canción.

<b>Procedure</b>	USP_ADD_TRACK_ARTIST	
Descripción	Asocia una canción con un artista.	
Retorno	XML	
Parámetros		
@TRACK_ID	int	ID de la canción en base de datos.
@ARTIST_ID	int	ID del artista en base de datos.

<b>Procedure</b>	USP_CLEAN_DB	
Descripción	Limpia la base de datos de toda información y deja únicamente los valores por defecto.	
Retorno	Ninguno	
Parámetros		
Ninguno		

<b>Procedure</b>	USP_GET_ALL_ALBUMS	
Descripción	Obtiene el listado de todos los álbumes en forma de entidad serializada en XML compatible con el modelo de clases C# del sistema.	
Retorno	XML	
Parámetros		
Ninguno		

<b>Procedure</b>	USP_GET_ALL_MESSAGES	
Descripción	Obtiene un listado de todas las cadenas de texto usadas en el sistema.	
Retorno	XML	
Parámetros		
Ninguno		

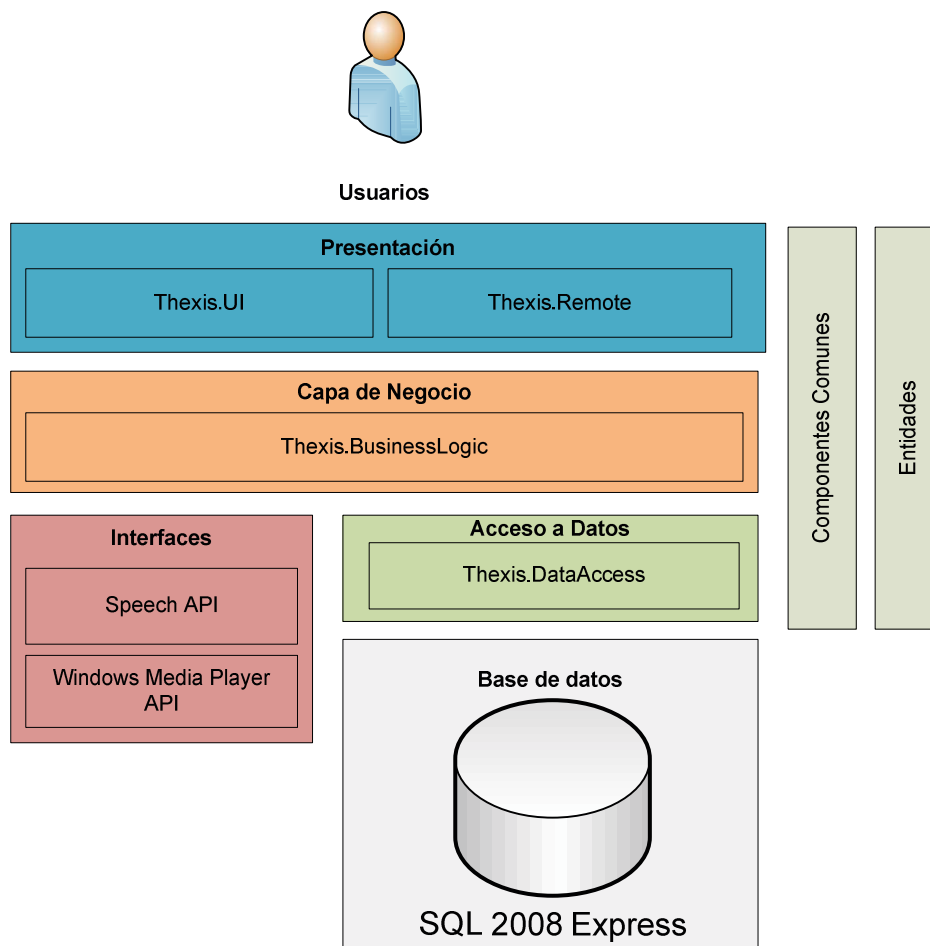
<b>Procedure</b>	USP_GET_LAMP_DEVICES	
Descripción	Obtiene el listado de todos los dispositivos de control de hogar inteligente en forma de entidad serializada en XML compatible con el modelo de clases C# del sistema.	
Retorno	XML	
Parámetros		
Ninguno		

<b>Procedure</b>	USP_INSERT_LOG	
Descripción	Inserta una entrada en el log de eventos del sistema.	
Retorno	XML	
Parámetros		
@LOG_DATE	date	Fecha del evento.
@LOG_DATE_TIME	datetime	Fecha y hora del evento.
@LOG_EVENT_TYPE	char(1)	Tipo de evento.
@LOG_MESSAGE	varchar(100)	Mensaje del evento.
@LOG_ORIGIN	varchar(100)	Origen del evento.
@LOG_STACK_TRACE	varchar(max)	Mensaje de tallado del evento.
@LOG_IP_ADDRESS	varchar(16)	Dirección IP asociada al evento.

<b>Procedure</b>	USP_GET_OBSERVABLE_DIRECTORIES	
Descripción	Obtiene un listado de URLs de directorios que el sistema debe monitorear por archivos multimedia.	
Retorno	XML	
Parámetros		
Ninguno		

### 3.2.2. Diseño de la Arquitectura Lógica

Imagen 3.9 Arquitectura lógica de la aplicación.



Fuente: Autor

La arquitectura lógica consta de las siguientes capas:

**Capa de presentación:** un programa elaborado en *Windows Forms*, y un *XML Web Services*, para control remoto de las funcionalidades del televisor, desde clientes de la intranet. Esta capa consume la capa de negocios, la capa de entidades y la de componentes comunes.

**Capa de negocio:** una librería de clases que implementa todas las funcionalidades del sistema. Consume la capa de entidades, la capa común, la capa de acceso a datos y las *APIs* de reconocimiento de voz y de *Windows Media Player*.

**Capa de acceso a datos:** una librería de clases cuya función es proveer de mecanismos para leer y grabar en la base de datos de la aplicación. Consume la capa de entidades, la capa común, y esta implementada para SQL Server 2008.

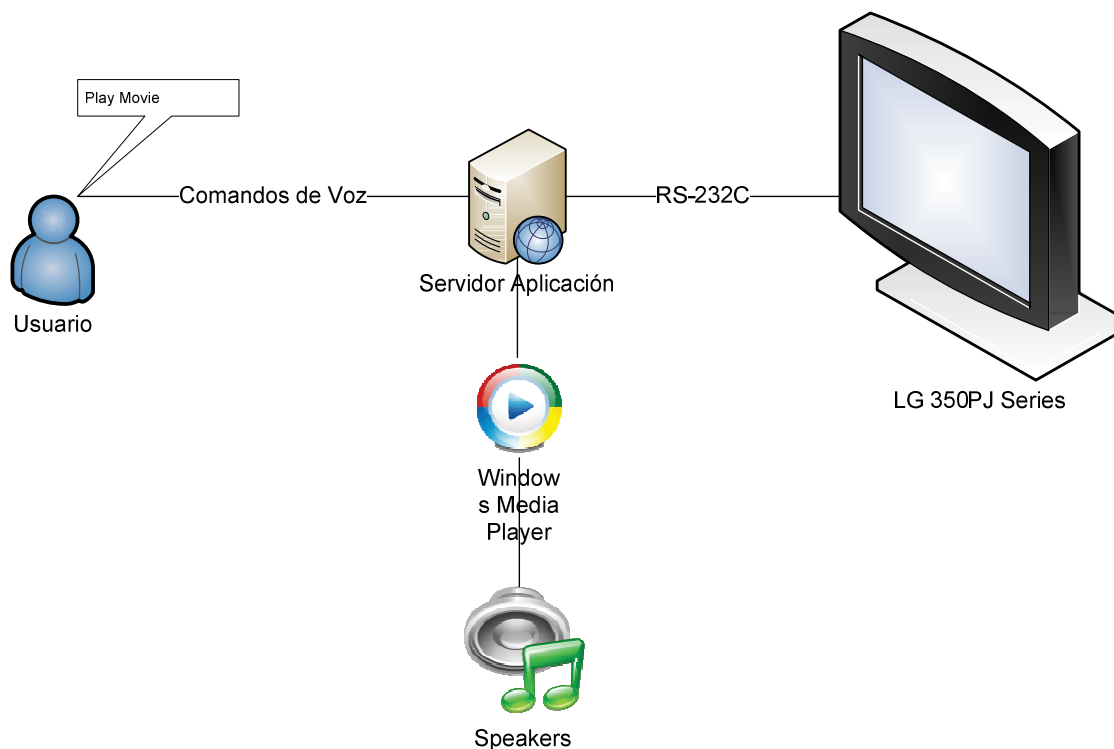
**Capa de entidades:** una librería de clases que contiene la definición de las entidades que modelan el sistema. Esta capa es común a todas las capas.

**Capa común:** una librería de clases que contiene funcionalidad genérica que es usada en todo el sistema.

**Capa de Interfaces:** contiene referencias a las APIs de programación para reconocimiento de voz y para el control del Windows Media Player.

### 3.2.3. Diseño de la Arquitectura Física

Imagen 3.10 Arquitectura física de la aplicación.



**Fuente:** Autor

La arquitectura física de la aplicación consiste de: un servidor de aplicaciones, que ejecutará el programa, y se conectará al televisor vía cable RGB o HDMI para la visualización de video, y a través del puerto RS232 para transmisión de comandos.

Televisor Plasma/LCD, que será donde se visualizará la pantalla de la aplicación y se podrá reproducir los videos elegidos por el usuario.

Parlantes que son por donde el sistema reproducirá los archivos de música y el audio de los videos.

### 3.2.4. Diseño de la Interfaz de Usuario

El diseño de la interfaz de usuario es minimalista, ya que la robustez del sistema radica en que se van a utilizar comandos de voz para controlar las funciones del software, dejando a la interfaz grafica las tareas menos comunes que comprenden, las pantallas de administración y de configuración.

A continuación se listan las pantallas del sistema, la función que desempeñarán dentro del sistema. Se detallan también, a un alto nivel, los elementos que va a contener cada pantalla del sistema.

#### **Pantalla de Carga del Sistema (*splash screen*).**

Imagen 3.11 Pantalla de carga del sistema.



Fuente: Autor

El propósito de esta pantalla es presentar una imagen de carga del sistema inmediatamente al usuario, para que éste tenga la impresión de que la respuesta del sistema es rápida. Mientras esta pantalla esta mostrándose, se van a consultar tablas en la base de datos, se va a inicializar también los componentes del sistema como son: motor de SAPI, reproductor de Windows media y por último, se construyen también las gramáticas del sistema.



## Pantalla Principal.

Esta será la interfaz principal del sistema, donde se le notificará al usuario que ya puede usar el programa. Las funciones de esta pantalla son las de capturar los comandos del usuario vía comandos de voz, y mostrar retroalimentación tanto visual como auditiva al usuario, sobre la correcta interpretación y ejecución de sus órdenes.

Esta pantalla contará con los siguientes elementos:

**Imagen 3.12** Pantalla principal del sistema



**Fuente:** Autor

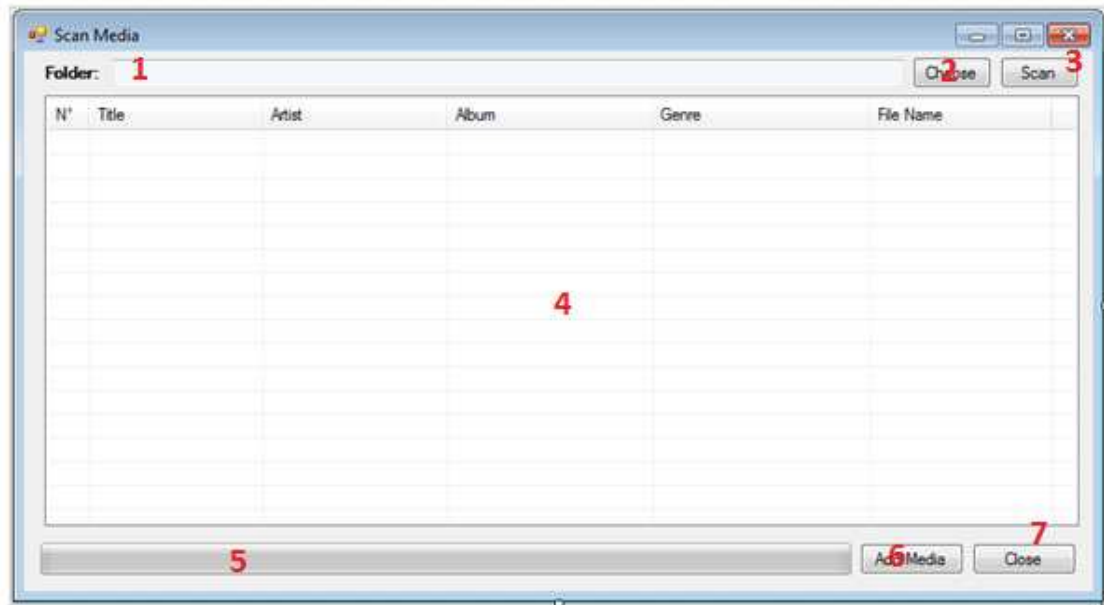
1. Reproductor de música
2. Control de habilitación / desactivación de reconocimiento de voz.
3. Botón de biblioteca multimedia.
4. Botón de configuración avanzada.
5. Área de información del contenido multimedia.
6. Área de estatus del sistema.

## Pantalla de Configuración de la Biblioteca Multimedia.

Esta pantalla de administración cumplirá con la función de dar mantenimiento a la biblioteca multimedia del sistema. Aquí el usuario podrá agregar archivos de música a la colección multimedia del sistema y también, realizar eliminación de los registros multimedia que ya no desee tener en el sistema.

Esta pantalla contará con los siguientes elementos:

**Imagen 3.13** Pantalla de configuración de la biblioteca multimedia.



**Fuente:** Autor

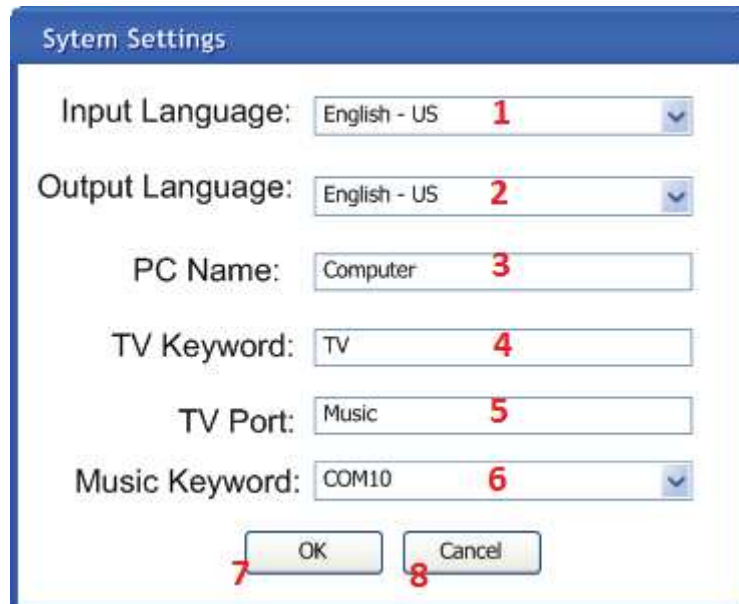
1. Ruta donde se están buscando los archivos multimedia.
2. Botón para escoger el directorio de búsqueda.
3. Botón que inicia la búsqueda de archivos multimedia.
4. Pantalla con los resultados de la búsqueda de los archivos multimedia.
5. Barra que indica el progreso de las operaciones de la pantalla.
6. Botón que graba los archivos seleccionados en el área (4) de la pantalla en la base de datos multimedia del sistema.
7. Botón de salir de este menú.

### **Pantalla de Configuración Avanzada**

Esta pantalla de administración tiene la función de permitirle al usuario configurar las opciones avanzadas del sistema, como son, idioma de reconocimiento de voz, idioma de conversión de texto a voz, número de cada uno de los puertos COM que utiliza el sistema.

Esta pantalla contará con los siguientes elementos:

**Imagen 3.14** Pantalla de configuración avanzada.



**Fuente:** Autor

1. Selección de Idioma de Entrada. Es posible activar el reconocimiento de voz en varios idiomas, dependiendo de si el sistema operativo soporta ese idioma.
2. Selección del Idioma de Salida. Aquí se configura en qué idioma se desea que el computador nos “hable”. También si un específico idioma esta soportado o no, dependerá del sistema operativo, pero se pueden instalar voces en idiomas adicionales provistas por terceros, por ejemplo: *Cepstral Voices*<sup>16</sup>.
3. Cuadro de configuración para el nombre del PC.
4. Cuadro de configuración para la palabra clave de los comandos de control de TV.
5. Cuadro de configuración para la palabra clave de los comando de control de música.
6. Selector de puerto COM en donde se ha conectado la TV.
7. Botón de OK. Graba la configuración en el sistema.
8. Botón de Cancelar. Cierra esta pantalla, y no graba ningún cambio a la configuración del sistema.

---

<sup>16</sup> Cepstral Voices, es un conjunto de voces compatibles con la especificación de SAPI 5.0 desarrollados por la compañía Cepstral Inc. Estas voces pueden ser usadas desde cualquier aplicación compatible con SAPI. Para más información visitar: [www.cepstral.com](http://www.cepstral.com).

### 3.2.5. Diseño de la Gramática del Sistema

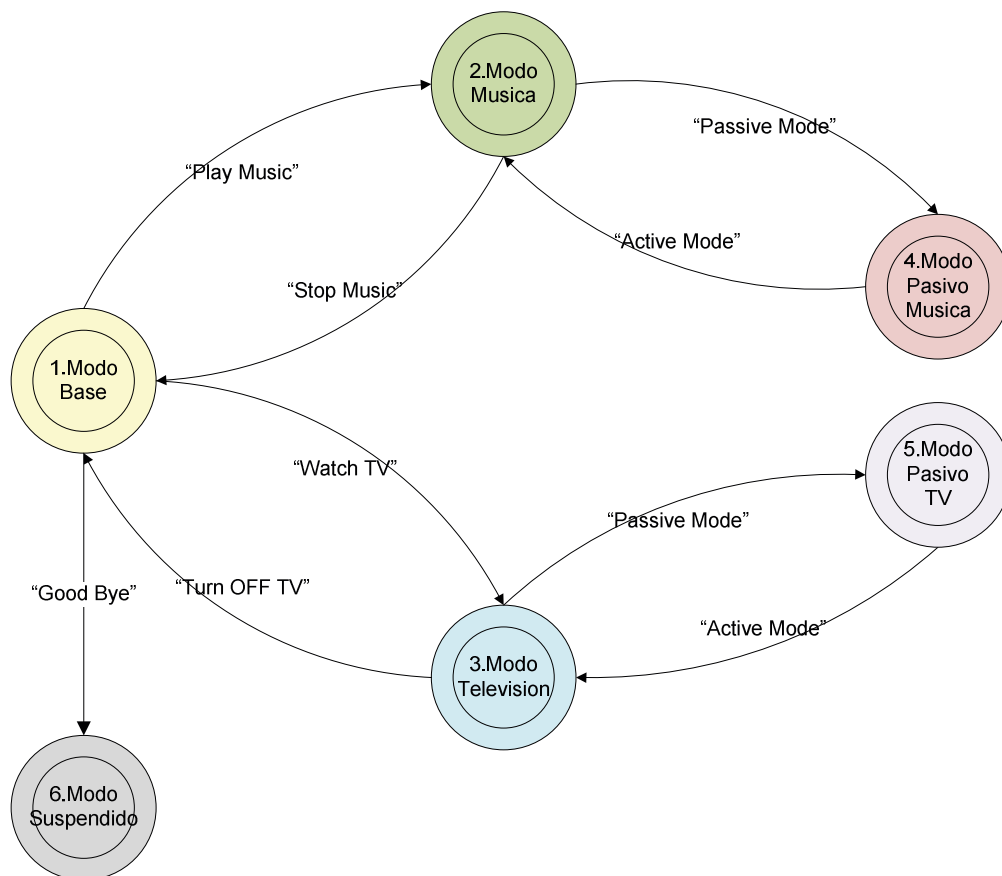
La gramática del sistema es una combinación de gramáticas especializadas en cada una de las tareas del sistema.

Estos grupos de tareas se pueden dividir en las siguientes categorías:

- Control del sistema base.
- Control de las funciones de televisión.
- Control de las funciones de audio.

Las cuales se activan según el usuario las necesite. El objetivo principal de esto es reducir la carga de trabajo del motor de reconocimiento de voz e incrementar la tasa de éxito de reconocimiento. Esto se debe a que el sistema de reconocimiento funciona de manera probabilística, asignando un peso y un valor de probabilidad a cada ítem que conforma la gramática. La relación entre número de ítems y probabilidad de reconocimiento de cada ítem es inversamente proporcional, es decir, a más ítems, es menor la probabilidad de reconocer correctamente una palabra. La siguiente imagen describe este comportamiento de la aplicación como una máquina de estados finitos.

Imagen 3.15 Máquina de Estados finito de la aplicación.



Fuente: Autor

A continuación se describen cada uno de los comandos que contendrán las gramáticas de los 3 modos principales del sistema.

### Modo Base:

Este será el estado inicial de la aplicación, los comandos que el usuario puede decir son:

- *Play Music*
- *Watch TV*
- *Goodbye*

La siguiente tabla resume el universo de comandos que el usuario puede decir en este modo:

**Tabla 3.1** Comandos de voz del sistema en modo base

Funcionalidad	Sintaxis del Comando
Iniciar el modo de música	[Computer]... Play Music
Iniciar el modo de TV	[Computer]... Watch TV
Salir del sistema	[Computer]... Goodbye

Fuente: Autor

### Modo Música:

Cuando el usuario se encuentra en el modo base y pronuncia el comando “*Play Music*”, se activa el modo de música en el sistema y la gramática del modo base se desactiva.

En este modo de la aplicación, los comandos que el usuario puede decir son:

- Play
- Pause
- Stop
- Play Artist
- Play Album
- Play Genre
- Play Track

La siguiente tabla resume el universo de comandos que el usuario puede decir en este modo:

**Tabla 3.2** Comandos de voz del sistema en modo música.

Funcionalidad	Sintaxis del Comando
Tocar / Reanudar la canción que se encuentra activa en el reproductor de música.	[Music]... Play
Pausar la reproducción de música.	[Music]... Pause
Detener la reproducción de música y volver al modo base.	[Music]... Stop
Tocar solamente las canciones que pertenecen a un solo artista en especial.	[Music]... Play Artist [Nombre del Artista]
Tocar solamente las canciones que pertenecen a un álbum en especial.	[Music]... Play Album [Nombre del Album]
Tocar todas las canciones que corresponden a un género musical.	[Music]... Play Genre [Nombre del Género]
Tocar una canción en especial.	[Music]... Play Track [Nombre de la Canción]

**Fuente:** Autor

### **Modo Televisión:**

Cuando el usuario se encuentra en el modo base y pronuncia el comando “*Watch TV*”, se activa el modo de televisión en el sistema y la gramática del modo base se desactiva.

En este modo de la aplicación, los comandos que el usuario puede decir son:

- Turn On TV
- Turn Off TV
- Mute
- Unmute
- Volume
- Channel
- Show Input

La siguiente tabla resume el universo de comandos que el usuario puede decir en este modo:

**Tabla 3.3** Comandos de voz del sistema en modo televisión.

Funcionalidad	Sintaxis del Comando
Encender la televisión.	[Computer]... Turn on TV
Apagar la televisión.	[Computer]... Turn off TV
Apagar el sonido de la TV.	[TV]... Mute
Restablecer el sonido de la TV.	[TV]... Unmute
Cambiar un canal.	[TV]... Channel [número]
Cambiar el volumen.	[TV]... Volume [número]
Seleccionar la entrada: (HDMI, TV, Video, etc.)	[TV]... Show [nombre de la entrada]

**Fuente:** Autor

Los distintos grupos de comandos serán agrupados en gramáticas, modeladas con las reglas disponibles en SAPI, y se procederá a activar y desactivar estas gramáticas conforme se las requiera en la aplicación.

### Integración con el motor de SAPI.

Ahora que tenemos diseñada la gramática del sistema, todo programador se preguntará, ¿cómo mi aplicación se va a enterar que se reconoció la voz del usuario en el sistema?

El reconocimiento de voz con SAPI sigue un patrón de programación denominado *producer / consumer*.

La aplicación instancia un *RecognitionContext*, al cual se le asigna una entrada de audio (en nuestro caso el micrófono) y una gramática (compuesta de reglas). Este contexto se convierte en el proveedor para nuestra aplicación.

La aplicación, al ser el consumidor, debe suscribirse a los eventos expuestos por el contexto a los cuales le interesa ser informado. En nuestro caso, debemos suscribirnos a estos eventos:

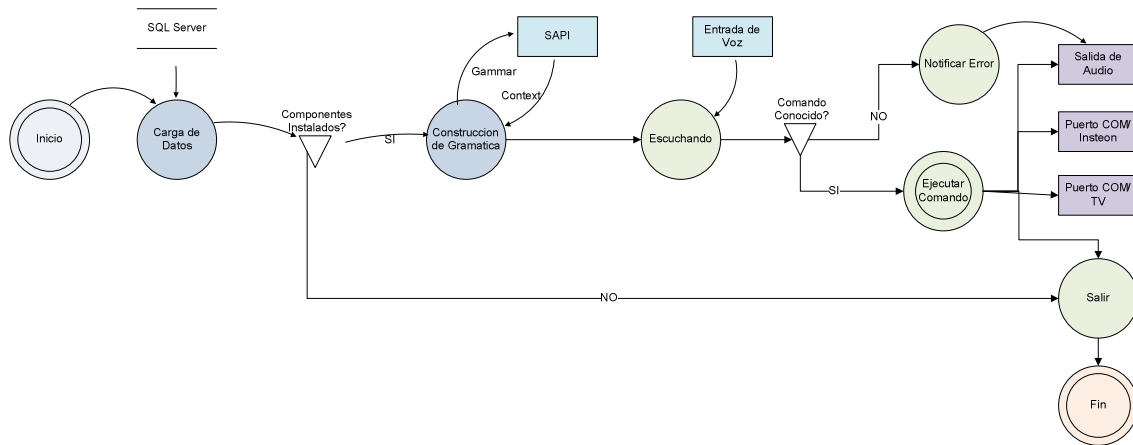
- Reconocimiento de Voz rechazado.
- Reconocimiento de Voz aceptado.

Una vez inicializado el contexto y suscrita la aplicación, se inicia el flujo de datos desde SAPI hasta nuestra aplicación. Básicamente lo que nos va a devolver SAPI en cada evento son tres cosas:

- El texto que se logró identificar
- La gramática y la regla donde se definió este texto y,
- Un número decimal entre 0 y 1 que indica el porcentaje de confiabilidad en los resultados.

Expresado gráficamente a un muy alto nivel, este será el flujo de datos de la aplicación, hasta llegar a la etapa de reconocimiento de voz.

**Imagen 3.16** Flujo de datos de la aplicación.



**Fuente:** Autor



## 4. Capítulo IV – Implementación y Pruebas

### 4.1. Implementación

#### Pruebas de concepto.

#### Control de la TV

Para diseñar e implementar la funcionalidad de control de TV, se recurrió al manual de usuario de los televisores LG, página 112-117 de la sección Apéndice, en donde se explican la serie de comandos que soporta la línea de televisores Plasma/LCD de este fabricante.

Los comandos a implementar son los siguientes:

- Encendido/Apagado
- Selección de fuente (AV/TV/Cable/PC/HDMI)
- Selección de canal.
- Control de volumen.

La forma en la que se implementan los comandos en los televisores LG es a través del puerto RS232, también conocido como puerto COM.

Los parámetros recomendados de configuración del puerto son los siguientes.

**Tabla 4.1** Parámetros de conexión para televisores LG.

Parámetro	Valor
Baud Rate	9600
Data Bits	8
Parity	None
Stop Bits	One
Encoding	ASCII

**Fuente:** Manual de Usuario de Televisores LCD y Plasma LG

A continuación, se explica que secuencia de caracteres se deben enviar al televisor para ejecutar las funciones que nuestra aplicación necesita.

### **Encendido:**

El encendido del televisor se logra enviando la cadena **ka 00 01** a través del puerto COM.

El televisor debe responder la cadena **a 01 OK01** en caso de que el comando sea ejecutado correctamente o la cadena **a 01 NG01**.

### **Apagado:**

El encendido del televisor se logra enviando la cadena **ka 00 00** a través del puerto COM.

El televisor debe responder la cadena **a 01 OK00** en caso de que el comando sea ejecutado correctamente o la cadena **a 01 NG00**.

### **Selección de Fuente:**

Los televisores con tecnología LCD o Plasma soportan varios formatos de entrada de video. Es común encontrarse con las siguientes entradas en un TV promedio:

- Análogo, sintonizador de TV, canales 1 – 125.
- Audio Video
- HDMI
- PC, también conocido como RGB
- Componente

Dependiendo del modelo del televisor, tendremos a disposición estos diferentes tipos de entrada.

La fórmula para el cálculo del valor hexadecimal que corresponde a cada es la siguiente:

Tipo de entrada + número de entrada – 1;

Los valores asignados a cada tipo de entrada se detallan en la siguiente tabla:

**Tabla 4.2** Valores asignados a cada entrada de los televisores LG

Entrada	Valor
Análogo (TV)	16
Audio Video	32
Componente	64
PC / RGB	96
HDMI	144

**Fuente:** Autor

La selección de la entrada del televisor se logra enviando la cadena **xb 00 ??** a través del puerto COM, donde **??** debe ser reemplazado por la conversión a hexadecimal de la fórmula: entrada + número de entrada - 1.

Por ejemplo, si queremos que el TV muestre la entrada HDMI1, debemos calcular la fórmula: 144 (HDMI) + 1 (número) - 1 = 144.

El comando a enviar será: **xb 00 90**

El televisor debe responder la cadena **b 01 OK90** en caso de que el comando sea ejecutado correctamente o la cadena **b 01 NG90**.

### Selección de Canal

La selección de canal en los televisores LG es una operación compleja ya que en la misma familia de televisores plasma, LCD y LED se hallan ciertos modelos que tienen incorporado un sintonizador de TV Digital, también conocida como DVB.

El implementar la sintonización por DVB representa un gran esfuerzo por lo cual se optó por usar el modelo de emulación de control remoto. Es decir, no se va a enviar el comando de sintonizador directamente al TV, en su lugar, se enviará la secuencia de botones que el usuario presionaría en su control remoto al momento de cambiar un canal.

La simulación de una tecla del control remoto del televisor se logra enviando la cadena **mc 00 ??** a través del puerto COM. Donde **??** es el valor de la tecla que se quiere simular, en hexadecimal.

La siguiente imagen resume los posibles valores de teclas de control remoto.

**Imagen 4.1** Comandos de control remoto de televisores LG.

Code (Hexa)	Function	Note
08	POWER	Remote control Button (Power On/Off)
45	Q.MENU	Remote control Button
43	MENU	Remote control Button
0B	INPUT	Remote control Button
10-19	Number Key 0-9	Remote control Button
09	MUTE	Remote control Button
02	VOL +	Remote control Button
03	VOL -	Remote control Button
00	CH ^	Remote control Button
01	CH v	Remote control Button
1E	FAV	Remote control Button
40	^	Remote control Button
41	v	Remote control Button
07	<	Remote control Button
06	>	Remote control Button
7C	X STUDIO	Remote control Button
44	ENTER	Remote control Button

**Fuente:** Manual de usuario de televisores LG.

Para realizar el sintonizado de canal se va a convertir a pulsaciones de teclas el número de canal deseado, y después, se van a enviar en secuencia, las pulsaciones que simulan la interacción por control remoto.

Por ejemplo, para seleccionar el canal 83, debemos enviar los comandos de control remoto: 8 – 3 – Enter.

Debemos enviar esta secuencia de instrucciones al puerto COM:

Tecla 8: **mc 00 18**

Tecla 3: **mc 00 13**

Tecla *Enter*: **mc 00 44**

El televisor debe responder la cadena **c 01 OK** en caso de que el comando sea ejecutado correctamente.

En caso de que un comando falle, se retornará **c 01 NG**.

### Control de Volumen:

El encendido del televisor se logra enviando la cadena **kf 00 ??** a través del puerto COM. Donde **??** debe ser reemplazado por el valor hexadecimal del volumen deseado. El valor del volumen debe ir en el rango de 0 a 100.

El televisor debe responder la cadena **f 01 OK** en caso de que el comando sea ejecutado correctamente o la cadena **f 01 NG**

La siguiente tabla, resume el set de instrucciones anteriormente mencionado:

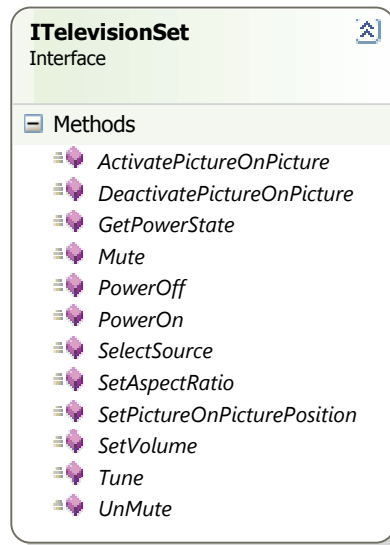
**Tabla 4.3** Comandos usados para control del televisor.

Función	Comando 1	Comando 2	DATA (Hex)
Encendido/Apagado	k	a	00 - 01
Selección de fuente (AV/Cable/HDMI, etc.)	x	b	Dato Complejo
Selección de canal.	m	c	Dato Complejo
Control de volumen.	k	f	00 - 64

**Fuente:** Autor

Se diseñó una interfaz llamada **ITelevisionSet** la cual declara formalmente el set básico de métodos que todos los televisores compatibles con la aplicación deben poder ofrecer.

Imagen 4.2 Interfaz ITelevisionSet



**Fuente:** Autor

La implementación de los métodos: *GetPowerState*, *Mute*, *PowerOff*, *PowerOn*, *SelectSource*, *SetVolume*, *Tune* y *UnMute* cubre el alcance definido en la sección 1.2 del presente documento.

La implementación de los métodos: *ActivatePictureOnPicture*, *DeactivatePictureOnPicture* y *SetAspectRatio*, no está contemplada dentro de los alcances del presente trabajo de investigación y no está garantizado que todos los televisores soporten este set de funciones, por lo que son considerados opcionales.

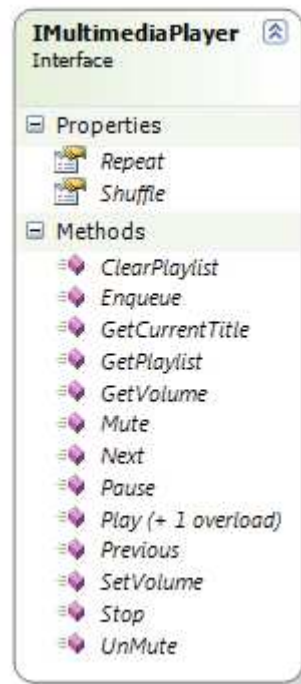
La creación de esta interfaz permite que a futuro, se puedan crear más implementaciones de la misma para diferentes marcas y modelos de televisores, respetando una abstracción. Esto, en programación orientada a objetos se lo conoce como polimorfismo.

Se creó una clase llamada LG30 que implementa esta interfaz, y que envía los comandos correspondientes en el puerto COM.

## Control de las funciones multimedia.

Primero se diseñó la interfaz *IMultimediaPlayer*, que expone los métodos descritos en la siguiente imagen.

Imagen 4.3 Interfaz IMultimediaPlayer



Fuente: Autor

Al igual que con la interfaz para controlar el televisor, el propósito de la interfaz *IMultimediaPlayer* es abstraer las operaciones del reproductor de música para que el sistema sea independiente de la implementación.

De esa manera, se podrían implementar clases que controlen diferentes tipos de reproductor de música, según sea la preferencia del usuario, sin realizar fuertes cambios en el sistema principal.

Se escogió implementar el API de Windows Media Player ya que este reproductor musical está presente en todas las versiones de Windows de escritorio.

Se obtuvo el API de Windows Media desde el sitio web de Microsoft. Este archivo lo conoce internamente Microsoft como Windows Media SDK. Actualmente se tiene disponible la versión 12 de este SDK, pero este archivo no se puede obtener independientemente, ya que forma parte del Platform SDK. Por simplicidad se recomienda trabajar con la versión 9 del SDK que si existe como una descarga independiente.

Este SDK expone un par de objetos COM+: WMPLib y WindowsMediaPlayer.

Se deben referenciar estos dos objetos COM+ en el proyecto de Visual Studio de donde se desee controlar el reproductor de Windows Media.

Se debe instanciar un objeto de la clase WindowsMediaPlayerClass el cual encapsula toda la función del reproductor de Windows media.

A partir de este objeto se implementan las operaciones de la interfaz IMultimediaPlayer.

En futuras implementaciones se puede crear nuevas implementaciones que contemplen otros reproductores de música, como Winamp, iTunes, Real Player, dependiendo de los requerimientos del usuario.

### **Control de dispositivos Insteon.**

Para controlar este tipo de dispositivos, se debe disponer del hardware adecuado, es decir, el módem que convertirá los comandos enviados por la aplicación en señales eléctricas que van a viajar por la red eléctrica del hogar.

Existe un SDK oficial por parte del fabricante, el cual lamentablemente es de acceso restringido y hay que pagar una cuota de inscripción en su programa de desarrolladores para poder obtener esta documentación y APIs oficiales.

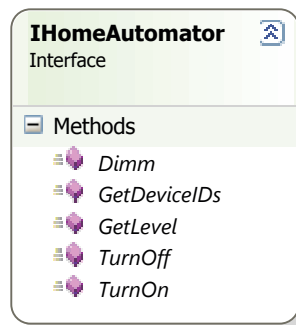


Sin embargo, existe en el internet proyectos de código abierto de APIs independientes para que las aplicaciones puedan enviar y recibir comandos Insteon.

Para el uso con C#, se recomienda la librería *FluentDwelling*, la cual es libre de uso y se lo distribuye bajo la modalidad de licencia MIT.

Se creó de igual forma, una interfaz, que encapsula el funcionamiento de un dispositivo de control de iluminación que soporta

**Imagen 4.4** Interfaz IHomeAutomator



**Fuente:** Autor

Para la implementación de esta interfaz, se creó una clase de negocio que hace uso de la librería *FluentDwelling* y que hace las llamadas API necesarias para poder llevar a cabo tareas como encendido, apagado y atenuación de Luces.

### **Conversión de Texto a Voz.**

Esta funcionalidad viene ya como una parte medular del Microsoft.NET Framework 3.5, por lo que no es necesario descargar ningún tipo de SDK.

Para implementar la conversión de texto a voz, se debe primero instanciar un sintetizador de voz. En la mayoría de versiones de Windows, viene instalado por defecto "Microsoft SAM" el cual será la voz y el personaje que convertirá a señal auditiva el texto que nuestra aplicación desee leer.

En Windows Vista y Windows 7, se cambia a Microsoft SAM por una versión nueva y mejorada versión llamada Microsoft Mary.

También, es posible instalar más de un sintetizador de voz en el sistema operativo, dotando de una amplia variedad de lenguajes, dialectos y personalidades a nuestro software.

La aplicación debe primero decidir que sintetizador se va a utilizar y se lo instancia. Después se envía una cadena de texto al sintetizador para que la lea. El sintetizador reproducirá por el dispositivo de audio por defecto, los parlantes, el texto convertido a la voz que el sintetizador genera.

### **Reconocimiento de Voz.**

Para realizar la tarea de reconocimiento de voz, se utilizó el motor de reconocimiento de voz provisto por el sistema operativo Windows.

Según la versión del sistema operativo, cambiará la versión del motor de reconocimiento de voz.

En Windows XP, el motor de reconocimiento de voz no está instalado por defecto en el sistema. Para poder contar con este componente se debe descargar el Microsoft Speech API (conocido también como SAPI) versión 5.1 o superior. También se puede conseguir el motor de reconocimiento de voz, instalando Microsoft Office 2003 / 2007 y habilitando la opción de dictado por voz.

### **Primer Prototipo**

El primer prototipo consta de los siguientes componentes: interfaz de usuario desarrollada en Windows Forms, la cual tiene botones que simulan los comandos de voz que el usuario dicta.

La aplicación contiene ya referencia a la capa de negocio en donde se han implementado las clases que controlan el televisor, los dispositivos Insteon y el reproductor de música.

### **Segundo Prototipo**

El segundo prototipo es una continuación del primer prototipo, en la cual en esta etapa se incluyen las funcionalidades de reconocimiento de voz y de conversión de texto a voz.

### **Versión Final**

La versión final parte del segundo prototipo, en el cual se incluye la capacidad de leer y almacenar información en la base de datos de SQL Server, lo que elimina los valores en duro en el aplicativo.

## 4.2. Pruebas

### 4.2.1. Pruebas Unitarias

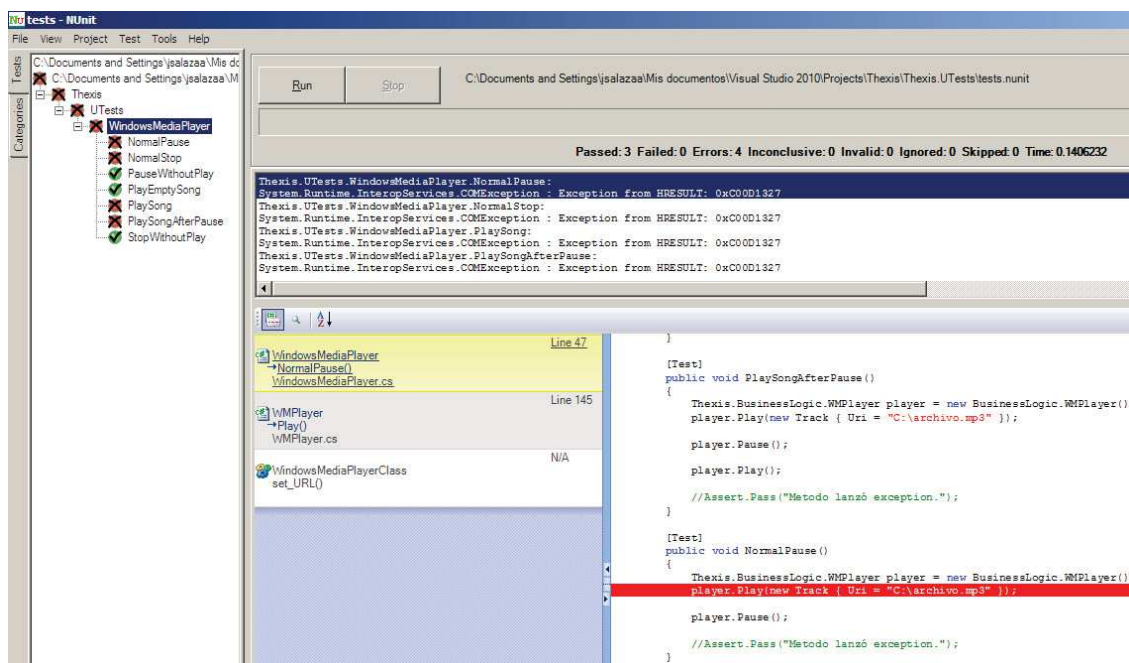
El propósito de las pruebas unitarias del sistema es el de asegurar que los componentes de bajo nivel del sistema funcionen según las especificaciones técnicas. En esta tarea, que se la debe realizar de la mano con la tarea de programación, el programador ejecuta llamadas a los métodos de bajo nivel de las clases de negocio, entidades, utilitarios, etc. y verifica que el método produce el efecto y la salida correctos.

Se decidió probar los componentes donde hubo más trabajo de implementación por parte del programador: el reproductor de música y el control de la televisión.

Para realizar estas pruebas se utilizó el utilitario para pruebas unitarias de código abierto llamado NUnit, versión 2.5.

Este utilitario ejecutara una serie de métodos programados (los que modelan una prueba), comparar si el resultado que arroja la ejecución es exactamente igual al resultado esperado de la prueba y mostrará de manera gráfica el éxito o fracaso de cada una de las pruebas. Color verde significa éxito, color rojo significa fracaso.

Imagen 4.5 Pantalla de resultados de NUnit.



Fuente: Captura de pantalla del programa NUnit.

El criterio de aceptación para certificar cada componente será:

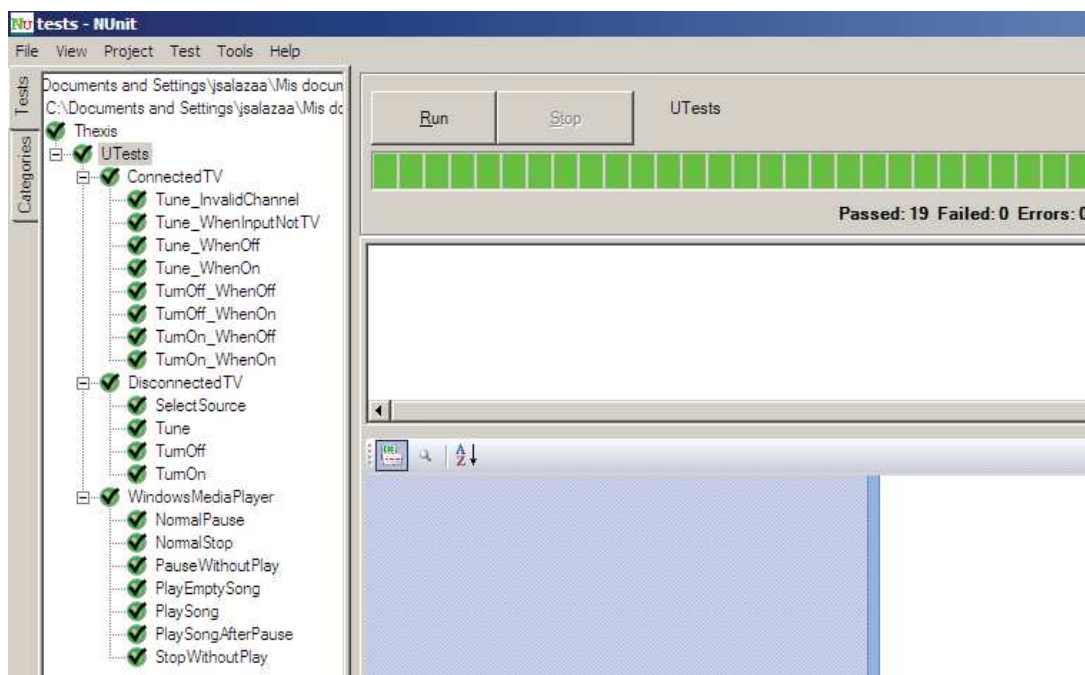
“Todas las pruebas del componente deben resultar en un 100% de casos de éxito”.

Se creó como parte de la solución de Visual Studio un proyecto más cuya función es contener únicamente las pruebas unitarias del sistema. En este se crearon varias clases, cada una contiene las diferentes pruebas programadas para los componentes en mención, TV y reproductor de música.

## Resultados de las pruebas

Después de ejecutar varios ciclos de pruebas y de realizar las correcciones del caso, se logó el 100% de casos de éxito, evidenciado por la herramienta NUnit.

**Imagen 4.6** Resultado de las pruebas unitarias del sistema.



**Fuente:** Herramienta NUnit.

A continuación se entra en detalle de cuál fue el criterio escogido en cada prueba, y las condiciones esperadas de salida de la ejecución de la prueba. Cada uno en una tabla para mayor facilidad de lectura.

## Reproductor de Windows Media.

Se probó la implementación de `IMultimediaPlayer`, programada en la clase `WMPlayer.cs`. Para cada método de negocio se programó una prueba con el *framework* de NUnit.

Método:	Play()
Descripción de la prueba	Invocar play sin haber seleccionado ninguna canción.
Resultado esperado	Excepcion
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	Play()
Descripción de la prueba	Invocar play habiendo seleccionado una canción.
Resultado esperado	La canción empieza a reproducirse.
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	Play()
Descripción de la prueba	Invocar play cuando se ha pausado una canción.
Resultado esperado	La canción se reproduce desde el punto donde se había pausado anteriormente.
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	Pause()
Descripción de la prueba	Invocar pause sin haber seleccionado ninguna canción.
Resultado esperado	Excepción
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	Pause()
Descripción de la prueba	Invocar pause con una canción reproduciéndose.
Resultado esperado	La música se detiene.
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	Stop()
Descripción de la prueba	Invocar stop sin haber eleccionado ninguna canción.
Resultado esperado	Nada
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	Stop()
Descripción de la prueba	Invocar stop con una canción en reproducción.
Resultado esperado	La música se detiene
Estado	EXITOSO
Observaciones	
Ninguno	

## Televisor

Se probó la implementación de ITelevisionSet, programada en la clase LG30R.cs. Para cada método de negocio se programó una prueba con el *framework* de NUnit

Métodos:	TurnOn(), TurnOff(), Tune(), SelectSource()
Descripción de la prueba	Invocar el metodo sin tener conexión.
Resultado esperado	Excepción de conexión.
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	TurnOn()
Descripción de la prueba	Invocar el método con el televisor encendido.
Resultado esperado	Mensaje de éxito.
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	TurnOn()
Descripción de la prueba	Invocar el método con el televisor apagado.
Resultado esperado	El televisor se enciende
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	TurnOff()
Descripción de la prueba	Invocar el método con el televisor apagado.
Resultado esperado	Mensaje de éxito.
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	TurnOff()
Descripción de la prueba	Invocar el método con el televisor encendido.
Resultado esperado	El televisor se apaga.
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	Tune()
Descripción de la prueba	Invocar el método con el televisor apagado.
Resultado esperado	Excepción.
Estado	EXITOSO
Observaciones	
Ninguno	



Método:	Tune()
Descripción de la prueba	Invocar el método con el televisor encendido y la fuente seleccionada es TV
Resultado esperado	Se cambia el canal.
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	Tune()
Descripción de la prueba	Invocar el método con el televisor encendido y la fuente seleccionada es diferente a TV.
Resultado esperado	Excepción.
Estado	EXITOSO
Observaciones	
Ninguno	

Método:	Tune()
Descripción de la prueba	Invocar el método con el televisor encendido y la fuente seleccionada es TV, pero invocando un número de canal invalido, superior a 125.
Resultado esperado	Excepción.
Estado	EXITOSO
Observaciones	
Ninguno	

#### 4.2.2. Pruebas del Sistema

En esta etapa del ciclo de vida del proyecto de desarrollo de software, se prueba funcionalmente el software, es decir, uno a uno los requerimientos funcionales son validados para comprobar que han sido satisfechos.

#### Estrategia de pruebas

Para probar el sistema se procedió a ejecutar uno a uno los requisitos funcionales descritos en la sección 2.2 del documento de especificaciones funcionales del sistema.

#### Criterio de Aceptación

El criterio de aceptación es el 80% de los casos de prueba que dependan de reconocimiento de voz, ejecutados exitosamente. Ya que la sección 2.3.1 del documento de Especificaciones Funcionales establece 80% como el promedio aceptable para el reconocimiento de comandos de voz.

Para las funciones que no dependen de ingreso de voz por parte del usuario, el 100% de los casos de pruebas deben ejecutarse correctamente.

### **Ambiente de Pruebas**

Para las pruebas del sistema se contó con el siguiente entorno:

#### **Computador**

DELL Inspiron 15R (laptop) con las siguientes características:

- Procesador Intel Core i3 2.4 Ghz, 4 núcleos.
- Memoria: 8GB de memoria DDR3
- Almacenamiento: 500 GB de disco duro.
- 3 Puertos USB, 1 Puerto eSATA
- 1 salida RGB
- 1 salida HDMI

#### **Software**

Se probó el sistema bajo la siguiente configuración de software:

- Sistema Operativo: Microsoft Windows 7 Ultimate 64bit Inglés (USA)
- Microsoft .NET Framework 3.5 Service Pack 1 64 bit.
- Microsoft SAPI 5.4 (viene como parte de Windows 7)
- Windows Media Player 12 (viene como parte de Windows 7)
- Windows Media Player SDK v10 32bit.
- Cepstral Voices v5.0 32 bit, se instalaron las voces de Cepstral Marta y Cepstral Miguel, que proveen al sistema de capacidad de hablar en español.
- Microsoft SQL Server 2008 Developer Edition Service Pack 1 64 bit.

#### **Televisor**

Se probó el sistema con un televisor Plasma LG 50PJ350R con las siguientes características:

- 50 Pulgadas
- 3 entradas HDMI
- 1 entrada USB
- 1 entrada RS232C (COM)
- 1 entrada RGP (PC)
- 2 entradas RCA (Audio Video)
- 2 entradas Component
- 1 entrada de Analog TV (antena y cable)

**Accesorios:**

Se usaron en las pruebas del sistema los siguientes accesorios:

- Microsoft XBOX 360 Wireless Receiver.
- Microsoft XBOX 360 Wireless Headset.
- Cable de conexión DB9 Male-Female (COM) de 1.5 metros de largo para conectar la television con la PC.
- Cable de conexión HDMI para conectar la television con la PC.

**Datos de Prueba**

Se usó una colección de 100 archivos de música de artistas variados para formar una colección multimedia con las siguientes características:

- Al menos 20 artistas diferentes.
- Al menos 20 álbumes diferentes.
- Al menos 5 géneros musicales diferentes.
- Que en la colección existan 2 canciones con el mismo nombre pero que correspondan a artistas diferentes.
- Que los nombres de las canciones y los nombres de los artistas correspondan al idioma Inglés.

**Medio Ambiente**

Se estableció un ambiente de pruebas adecuado, una sala de estar en un apartamento promedio, donde la televisión y el equipo de cómputo puedan estar conectados a una distancia prudente. El usuario se ubicó a 5 metros de distancia de la televisión gracias a que el micrófono elegido es un micrófono inalámbrico. El volumen de la música y del televisor se configuró en 15% de capacidad, un volumen aceptable para un ambiente de ruido normal.

**Resultados de las Pruebas**

Los resultados muestran un correcto comportamiento del sistema de acuerdo a los parámetros establecidos en el documento de especificaciones funcionales.

Se nota una baja notable en la precisión del reconocimiento de voz cuando las condiciones acústicas no son favorables, por ejemplo, cuando el volumen de la televisión o de la música está por encima del 50%.

Si se incrementa el volumen de la televisión y de la música por encima del 70%, no se pueden obtener casos de éxito en los casos de pruebas.

## 5. Capítulo V – Conclusiones y Recomendaciones

### 5.1. Conclusiones

Del presente trabajo de tesis, el autor puede concluir:

1. El control de dispositivos del hogar es una aplicación completamente viable para la tecnología de reconocimiento de voz, ya que existen las facilidades tanto desde la perspectiva del software como también desde la perspectiva del hardware.
2. El sistema desarrollado se complementa de manera adecuada con el hardware al cual estaba destinado, en este caso la serie de televisores LG 30R, y con la plataforma de software para la cual se ha diseñado, en este caso el sistema operativo Windows, gracias a que existe la documentación y las herramientas de software adecuadas y están a disposición del público.
3. Los objetivos propuestos en este trabajo de tesis se han cubierto de manera exitosa gracias a que la tarea de desarrollo de software se ha complementado y reforzado con sólidas bases teóricas, resultado de la investigación realizada sobre los principios detrás del reconocimiento de voz.
4. Se puede concluir también que en el futuro cercano, la tendencia de los fabricantes de aparatos destinados al hogar será la de crear dispositivos cada vez más inteligentes, que se integren a la red doméstica del hogar y que puedan comunicarse con los demás aparatos inteligentes de manera cooperativa, lo cual hace necesario el surgimiento de nuevos protocolos de comunicación y nuevos programas de software que ayuden al usuario doméstico a controlar todo este ecosistema de dispositivos de una manera más amigable y natural, para lo cual, las interfaces de usuario naturales (NUI: *Natural User Interface*), entre éstas, el reconocimiento de voz, el reconocimiento de imágenes y de gestos, serán la apuesta de la industria del software y estas características estarán incluidas de manera nativa en futuros sistemas operativos y aparatos domésticos.
5. Por último, se debe resaltar también el gran potencial de la tecnología de síntesis de voz (*Text to Speech*) la cual es relativamente más fácil de implementar que el reconocimiento de voz; razón por la cual va a estar presente en muchos más dispositivos. Actualmente se puede encontrar esta funcionalidad en variedad de dispositivos electrónicos, desde teléfonos celulares, *iPods*, así como en aplicaciones más serias, para nombrar unas cuantas: dispositivos GPS que asisten al usuario a encontrar su camino, en *call centers* en donde se emiten comunicados automáticamente al usuario sin tener que contratar a una persona que

pre-grabe estos mensajes, e incluso en aviones comerciales en donde se notifica al piloto de alertas serias en las cuales una simple luz en el panel o un simple *beep* en la cabina de mando no serian suficiente para comunicar a un humano sobre la seriedad de un evento.

## 5.2. Recomendaciones

- Se recomienda siempre usar las más recientes versiones de los SDK para desarrollo de software, sobre todo cuando se desea crear aplicaciones para las últimas versiones de sistema operativo, por ejemplo Windows 7 o Windows 8, ya que no se garantiza la compatibilidad de anteriores SDK con las nuevas versiones de Windows. El camino inverso, es decir, usar el ultimo SDK para crear aplicaciones que apunten a plataformas antiguas si está soportado ya que por lo general existe “*backwards compatibility*” en cada nueva versión, es decir, las aplicaciones desarrolladas con los nuevos SDK’s podrían llegar a ser compatibles con las antiguas versiones de sistemas operativos.
- Se recomienda mantener el universo de palabras que el motor de reconocimiento de voz debe intentar identificar, en un máximo de 100 palabras diferentes por gramática. De esta manera se puede obtener una precisión aceptable del software de reconocimiento de voz. Si se requiere identificar un gran número de comandos diferentes, se recomienda separar los comandos en gramáticas diferentes y activar las gramáticas en la aplicación según sea necesario.
- Se recomienda ampliar las funcionalidades del sistema, para cubrir aún más tareas multimedia y de ocio en el hogar. Por ejemplo: se pueden implementar clases que consuman servicios en línea, como por ejemplo YouTube, Facebook, Twitter, UStream (por mencionar algunos cuantos). Estos sitios disponen de APIs de programación que el desarrollador podría consumir en la aplicación, y lograr una experiencia de usuario única. Algún día se podría llegar a tener este tipo de interacción en nuestros hogares: “Computadora, busca en YouTube videos de Nirvana” o “Computadora, muéstrame las fotos de Pepito Perez en Facebook” o “Computadora, avísame cuando haya aterrizado el vuelo LH540 que viene de Frankfurt”.
- Se recomienda también estudiar la factibilidad comercial de llegar a consumir desde este sistema, servicios Web empresariales, apalancados en una plataforma de pago de algún banco local, para poder realizar tareas de compras en línea. Un ejemplo teórico de este tipo de implementación del sistema sería: “Computadora, comprar una entrada para la película XYZ en Multicines para Hoy a las 9pm”.

- Se recomienda también adecuar el sistema para realizar pequeñas tareas específicas dentro de edificios inteligentes, por ejemplo: en un ascensor se podría usar una gramática simple para reconocer la voz del usuario y controlar por voz a qué piso se debe ir.
- Se recomienda implementar el sistema en casas inteligentes para ayudar a los usuarios con limitaciones físicas, donde el acceso a los controles físicos para la iluminación y/o el control de temperatura es limitado o de difícil operación para el usuario. Con este sistema, el usuario podría controlar estas funciones a través de su voz sin mayores complicaciones. Este uso también es aplicable a hospitales.

## 6. Bibliografía

### Libro:

Edgar, B. (2001). *The Voice XML Handbook: understanding and building the phone-enabled web*. CMP Books.

Kortum, P. (2008). *HCI Beyond the GUI: Design for Haptic, Speech, Olfactory and Other Nontraditional Interfaces*. Morgan Kaufmann Publishers.

Nas, C., & Brave, S. (2005). *Wired for Speech: How Voice Activates and Advances the Human-Computer Relationship*. The MIT Press.

Sugumaran, V. (2008). *Intelligent Information Technologies: Concepts, Methodologies, Tools, and Applications*. IGI Global.

Zurk, J. (2004). *Integration-ready architecture and design: software engineering with xml. java, .net, wireless, speech and knowledge technologies*. Cambridge University Press.

### Documento de Internet:

LG Electronics. (2011). *LG 50PJ350 Televisions - High Definition Plasma TV - Owner's Manual*. Recuperado el 18 de 03 de 2011, de <http://www.lg.com/common/product/support.jsp?localeCd=US&productId=50PJ350&uri=/us/tv-audio-video/televisions/LG-plasma-tv-50PJ350.jsp&categoryIdCms=3#>

Microsoft Corporation. (s.f.). *Developing for Speech - MSDN*. Recuperado el 27 de 02 de 2012, de <http://msdn.microsoft.com/en-us/library/bb756992.aspx>

Microsoft Corporation. (n.d.). *Language Integrated Query*. Retrieved from MSDN: <http://msdn.microsoft.com/en-us/library/bb397926.aspx>

Microsoft Corporation. *SAPI SDK Help*. Recuperado el 27 de 02 de 2012, de:

<http://www.microsoft.com/downloads/info.aspx?na=41&srcfamilyid=5e86ec97-40a7-453f-b0ee-6583171b4530&srcdisplaylang=en&u=http%3a%2f%2fdownload.microsoft.com%2fdownload%2fB%2f4%2f3%2fB4314928-7B71-4336-9DE7-6FA4CF00B7B3%2fsapi.chm>

Open Source Initiative. (s.f.). *opensource.org*. Recuperado el 26 de 02 de 2012, de <http://www.opensource.org/licenses/mit-license.php>

Sun Microsystems / Oracle. (s.f.). *K Virtual Machine*. Obtenido de Sun Developer Network: <http://java.sun.com/products/cldc/wp/>

Sun Microsystems. (s.f.). *JSAPI Introduction*. Obtenido de Sun Developer Network: [java.sun.com/products/java-media/speech/forDevelopers/jsapi-guide/Introduction.html](http://java.sun.com/products/java-media/speech/forDevelopers/jsapi-guide/Introduction.html)

Wikipedia. (s.f.). *ATL Active Template Library*. Obtenido de [http://en.wikipedia.org/wiki/Active\\_Template\\_Library](http://en.wikipedia.org/wiki/Active_Template_Library)

Wikipedia. (s.f.). *CSharp*. Obtenido de [http://en.wikipedia.org/wiki/C\\_Sharp\\_%28programming\\_language%29](http://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29)

Wikipedia. (s.f.). *Dalvik*. Obtenido de [http://en.wikipedia.org/wiki/Dalvik\\_%28software%29](http://en.wikipedia.org/wiki/Dalvik_%28software%29)

Wikipedia. (s.f.). *Dynamic Link Library*. Obtenido de [http://en.wikipedia.org/wiki/Dynamic-link\\_library](http://en.wikipedia.org/wiki/Dynamic-link_library)

Wikipedia. (n.d.). *Hidden Markov Model*. Recuperado el 02 27, 2012, de [http://en.wikipedia.org/wiki/Hidden\\_Markov\\_model](http://en.wikipedia.org/wiki/Hidden_Markov_model)

Wikipedia. (s.f.). *Java Native Interface*. Obtenido de [http://en.wikipedia.org/wiki/Java\\_Native\\_Interface](http://en.wikipedia.org/wiki/Java_Native_Interface)

Wikipedia. (s.f.). *Market Share*. Obtenido de Windows Mobile: [http://en.wikipedia.org/wiki/Windows\\_Mobile#Market\\_share](http://en.wikipedia.org/wiki/Windows_Mobile#Market_share)

Wikipedia. (s.f.). *Markov Process*. Recuperado el 27 de 02 de 2012, de [http://en.wikipedia.org/wiki/Markov\\_process](http://en.wikipedia.org/wiki/Markov_process)

Wikipedia. (n.d.). *Mel Scale*. Recuperado el 02 27, 2012, de [http://en.wikipedia.org/wiki/Mel\\_scale](http://en.wikipedia.org/wiki/Mel_scale)

Wikipedia. (n.d.). *Microsoft Speech API*. Recuperado el 12 24, 2011, de [http://en.wikipedia.org/wiki/Microsoft\\_Speech\\_API](http://en.wikipedia.org/wiki/Microsoft_Speech_API)

Wikipedia. (n.d.). *MIME*. Recuperado el 02 27, 2012, de <http://en.wikipedia.org/wiki/MIME>

Wikipedia. (s.f.). *Neural Network*. Recuperado el 27 de 02 de 2012, de [http://en.wikipedia.org/wiki/Neural\\_network](http://en.wikipedia.org/wiki/Neural_network)

Wikipedia. (s.f.). *Pulse Code Modulation (PCM)*. Recuperado el 27 de 02 de 2012, de [http://en.wikipedia.org/wiki/Pulse-code\\_modulation](http://en.wikipedia.org/wiki/Pulse-code_modulation)

Wikipedia. (s.f.). *Smartphone World Market Share*. Obtenido de [http://en.wikipedia.org/wiki/Mobile\\_operating\\_system](http://en.wikipedia.org/wiki/Mobile_operating_system)



Wikipedia. (s.f.). *Stochastic Process*. Recuperado el 27 de 02 de 2012, de [http://en.wikipedia.org/wiki/Stochastic\\_process](http://en.wikipedia.org/wiki/Stochastic_process)

Wikipedia. (2012, 01 29). *Windows 9X*. Recuperado el 02 23, 2012, de [http://en.wikipedia.org/wiki/Windows\\_9x](http://en.wikipedia.org/wiki/Windows_9x)

Wikipedia. (s.f.). *Windows Mobile*. Obtenido de Market Share: [http://en.wikipedia.org/wiki/Windows\\_Mobile#Market\\_share](http://en.wikipedia.org/wiki/Windows_Mobile#Market_share)

Wikipedia. (2012, 02 06). *Windows NT*. Recuperado el 02 23, 2012, de [http://en.wikipedia.org/wiki/Windows\\_NT](http://en.wikipedia.org/wiki/Windows_NT)

## 7. Anexos

### Anexo 1.

#### The MIT License (MIT)

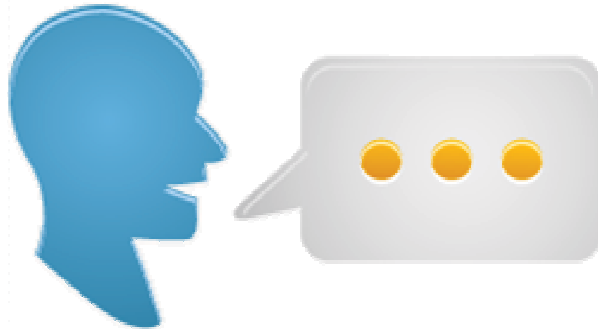
Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## **Anexo 2.**



# Speech Media Controller

Sistema de Control por Comandos de Voz de las Funciones  
Multimedia del Hogar

## Manual de Instalación v1.0

Mayo 2012

### Tabla de contenido

Requisitos de Software .....	2
Requisitos mínimos de Hardware .....	2
Requisitos recomendados de Hardware .....	2
Instalación del Sistema .....	3
Instalación de Microsoft .NET Framework .....	3
Instalación de SQL Server Express .....	3
Instalación de Microsoft Speech API .....	4
Instalación de Windows Media Player SDK .....	4
Instalación del software de control de tv y música .....	5

## Requisitos de Software

- El sistema de reconocimiento de voz para los dispositivos del hogar tiene los siguientes requisitos de software:
- Windows XP SP3, Windows Server 2003 SP1, Windows Vista, Windows Server 2008, Windows 7 o superior.
- SQL Server 2008 Express Edition.
- Windows Media Player 32 bit version 9 o superior.

## Requisitos mínimos de Hardware

- Procesador: mínimo 1Ghz, con arquitectura compatible con x86 o x64: Intel Pentium III, Pentium 4, Core 2 Duo, Xenon, Core i3, Core i5, Core i7, Atom, AMD Athlon, Athlon XP, Athlon 64, Turion, etc.
- Memoria: mínimo 1GB de memoria ram.
- Disco duro: 16 GB mínimo de espacio en disco para Windows y 1 GB adicional para las aplicaciones.
- Tarjeta de sonido de 32 bits.
- Micrófono y parlantes.
- Cable de conexión DB9 (puerto COM) male-female.
- Un puerto DB9 (puerto COM) disponible, o en su defecto se puede instalar un convertidor de puerto USB a COM.
- Televisor Plasma, LCD o LED compatible con la serie LG30R con entrada de puerto RS232 (puerto COM).

## Requisitos recomendados de Hardware

- Procesador: 2.4 Ghz, con arquitectura compatible con x86 o x64: Core 2 Duo, Xenon, Core i3, Core i5, Core i7.
- Memoria: 4GB de memoria ram.
- Disco duro: 20 GB mínimo de espacio en disco para Windows y 1 GB adicional para las aplicaciones.
- Tarjeta de sonido de 32 bits.
- Micrófono inalámbrico y parlantes tipo home theater.
- Cable de conexión DB9 (puerto COM) male-female.
- Un puerto DB9 (puerto COM) disponible, o en su defecto se puede instalar un convertidor de puerto USB a COM.
- Televisor Plasma, LCD o LED compatible con la serie LG30R con entrada de puerto RS232 (puerto COM).

## **Instalación del Sistema**

### **Instalación de Microsoft .NET Framework**

Para obtener este componente, visitar la dirección de internet:

<http://www.microsoft.com/download/en/details.aspx?id=25150>

Descargar el componente Microsoft.NET Framework 3.5 SP1, esta descarga tarda aproximadamente 5 minutos con una conexión de 512kbps. Una vez que termine la descarga, continuar con el siguiente paso.

#### **Instalación:**

Ejecutar el archivo “dotnetfx35.exe”.

Seguir las instrucciones que aparecen en pantalla.

### **Instalación de SQL Server Express**

Para obtener este componente, visitar la dirección de internet:

<http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=3743>

Descargar el componente SQLEXPRESS\_x86\_ENU.exe, esta descarga tarda aproximadamente 3 minutos con una conexión de 512kbps. Una vez que termine la descarga, continuar con el siguiente paso.

#### **Instalación:**

Ejecutar el archivo “SQLEXPRESS\_x86\_ENU.exe”.

Seguir las instrucciones que aparecen en la pantalla.

## **Instalación de Microsoft Speech API**

IMPORTANTE: Si tiene Windows Vista o Windows 7, por favor omitir este paso y continuar con el resto de la instalación.

Para obtener el componente, visitar la dirección de internet:

<http://www.microsoft.com/download/en/details.aspx?id=10121>

Descargar el componente SpeechSDK51.exe, esta descarga tarda aproximadamente 3 minutos con una conexión de 512kbps. Una vez que termine la descarga, continuar con el siguiente paso.

### **Instalación:**

Ejecutar el archivo "SpeechSDK51.exe".

Seguir las instrucciones que aparecen en pantalla.

## **Instalación de Windows Media Player SDK**

IMPORTANTE: Si tiene Windows XP Service Pack 3, Windows Vista o Windows 7, por favor omitir este paso y continuar con el resto de la instalación. En las versiones de Windows Server 2003 o Windows Server 2008 es necesario instalar Windows Media Player ya que no se encuentra instalado por defecto en el sistema operativo.

Para descargar Windows Media Player, por favor visitar la siguiente dirección de internet:

<http://windows.microsoft.com/en-US/windows/downloads/windows-media-player>

Elegir la versión de Windows Media Player que esta destinada a su versión de sistema operativo.

Para descargar Windows Media Player SDK, por favor visitar la siguiente dirección de internet:

<http://www.microsoft.com/download/en/details.aspx?id=25243>

Descargar el componente WMPlayerSDK.exe e instalarlo siguiendo las instrucciones que aparecen en pantalla.

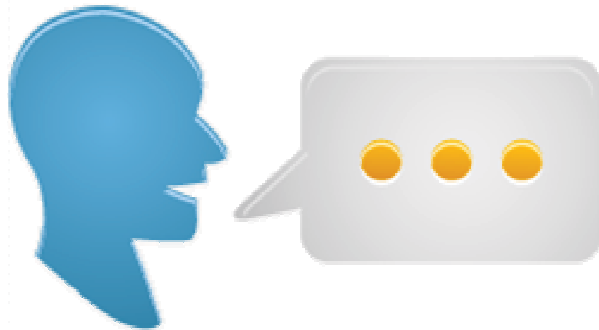
### **Instalación del software de control de tv y música.**

Ejecutar el archivo “setup.exe”.

Seguir las instrucciones que aparecen en pantalla.



### **Anexo 3.**



# **Speech Media Controller**

Sistema de Control por Comandos de Voz de las Funciones  
Multimedia del Hogar

**Manual de Usuario v1.0**

Mayo 2012

## Tabla de contenido

Requisitos del Sistema.....	3
Requisitos de Software .....	3
Requisitos mínimos de Hardware .....	3
Requisitos recomendados de Hardware .....	3
Ejecución del Programa Principal.....	4
Control de Funciones Básicas.....	6
Salir del sistema.....	6
Iniciar la televisión.....	6
Iniciar la música. ....	6
Desactivar micrófono.....	7
Activar micrófono. ....	7
Control de Televisión .....	7
Apagar. ....	7
Encender. ....	7
Silenciar la televisión.....	7
Quitar el silencio.....	7
Elegir el volumen.....	7
Cambiar el canal. ....	8
Cambiar la entrada de video.....	8
Control de Música .....	8
Reproducir música. ....	8
Pausar .....	8
Detener.....	8
Reanudar la reproducción. ....	9
Silenciar.....	9
Quitar el silencio .....	9
Elegir el volumen.....	9
Búsqueda de artista. ....	9
Búsqueda de álbum. ....	9
Búsqueda de canción.....	9
Administración de la Biblioteca Multimedia .....	10
Configuración avanzada del sistema .....	12
Solución a problemas frecuentes .....	13

## Requisitos del Sistema

### Requisitos de Software

El sistema de reconocimiento de voz para los dispositivos del hogar tiene los siguientes requisitos de software:

- Windows XP SP3, Windows Server 2003 SP1, Windows Vista, Windows Server 2008, Windows 7 o superior.
- SQL Server 2008 Express Edition.
- Windows Media Player 32 bit versión 9 o superior.

### Requisitos mínimos de Hardware

- Procesador: mínimo 1Ghz, con arquitectura compatible con x86 o x64: Intel Pentium III, Pentium 4, Core 2 Duo, Xenon, Core i3, Core i5, Core i7, Atom, AMD Athlon, Athlon XP, Athlon 64, Turion, etc.
- Memoria: mínimo 1GB de memoria ram.
- Disco duro: 16 GB mínimo de espacio en disco para Windows y 1 GB adicional para las aplicaciones.
- Tarjeta de sonido de 32 bits.
- Micrófono y parlantes.
- Cable de conexión DB9 (puerto COM) male-female.
- Un puerto DB9 (puerto COM) disponible, o en su defecto se puede instalar un convertidor de puerto USB a COM.
- Televisor Plasma, LCD o LED compatible con la serie LG30R con entrada de puerto RS232 (puerto COM).

### Requisitos recomendados de Hardware

- Procesador: 2.4 Ghz, con arquitectura compatible con x86 o x64: Core 2 Duo, Xenon, Core i3, Core i5, Core i7.
- Memoria: 4GB de memoria ram.
- Disco duro: 20 GB mínimo de espacio en disco para Windows y 1 GB adicional para las aplicaciones.
- Tarjeta de sonido de 32 bits.
- Micrófono inalámbrico y parlantes tipo home theater.
- Cable de conexión DB9 (puerto COM) male-female.
- Un puerto DB9 (puerto COM) disponible, o en su defecto se puede instalar un convertidor de puerto USB a COM.
- Televisor Plasma, LCD o LED compatible con la serie LG30R con entrada de puerto RS232 (puerto COM).

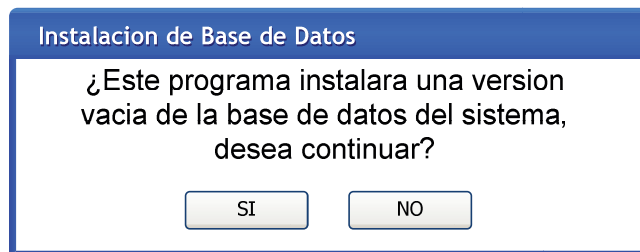
## Ejecución del Programa Principal

Desde el escritorio de Windows, dirigirse al menú:

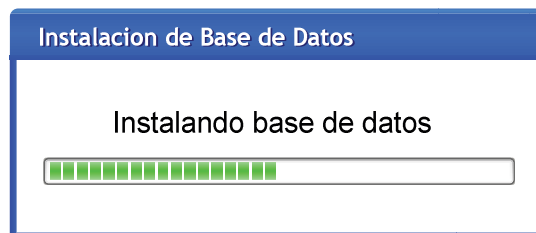
Inicio → Programas → Control de Audio y TV → Speech Media Controller

Espere a que el programa realice la carga inicial del sistema.

**IMPORTANTE:** Si es la primera vez que ejecuta el programa, se debe inicializar la base de datos del sistema. Debe aparecer esta pantalla. Se debe dar click en la opción "SI". Si este paso ya se ha ejecutado con anterioridad, es posible que no se pueda conectar con la base de datos del sistema. Por favor revise la documentación en la parte de solución de problemas.



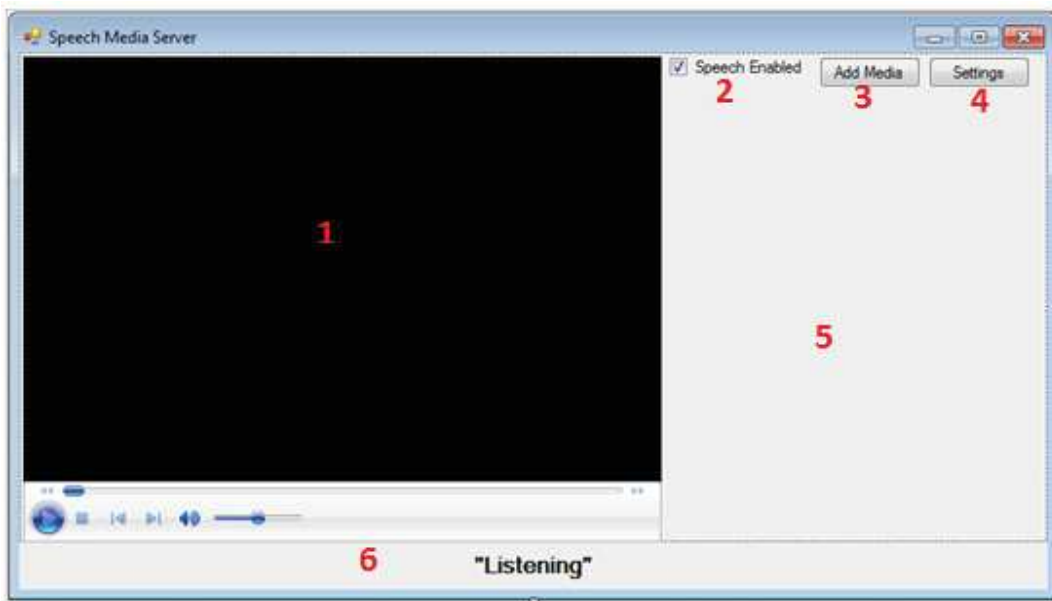
El asistente se tomara unos instantes para instalar la base de datos del sistema.



Una vez que se ejecute este asistente se continuará con la ejecución normal del programa.



Una vez que finalice la carga del software, se visualizará la pantalla principal del sistema.



Esta pantalla cuenta con los siguientes elementos:

1. Reproductor de música
2. Control de habilitación / desactivación de reconocimiento de voz.
3. Botón de biblioteca multimedia.
4. Botón de configuración avanzada.
5. Área de información del contenido multimedia.
6. Área de estatus del sistema.

Cada área cumple una función específica del sistema.

## Control de Funciones Básicas

En esta parte del manual no se van a incluir capturas de pantallas ya que el control del software se lo realiza por comandos de voz.

Para enviar un comando, se debe seguir siempre la sintaxis:

Nombre del PC... Comando.

La entonación debe ser normal, no muy rápida, no muy lenta, tampoco en un tono de voz muy débil ni demasiado fuerte. Tal vez un buen ejemplo de pronunciación sería, como lee un reportero de noticias del clima en un programa de TV.

El nombre del PC que se tiene configurado por defecto es "Computer" la cual será la palabra mágica con la cual la computadora sabrá que debe esperar una instrucción. De esta forma se evita el tener mal entendidos por parte del sistema ya que el micrófono podría captar todas las conversaciones de que se efectúen en sus cercanías.

Esta palabra clave se puede cambiar en el menú de configuración del sistema.

### Salir del sistema.

Comando: "Goodbye".

Ejemplo: diga al micrófono el comando "Computer"... Goodbye

### Iniciar la televisión

Comando: "Watch TV".

Ejemplo: diga al micrófono el comando "Computer"... "Watch TV".

Una vez que se haya activado el modo de televisión se podrán ejecutar los comandos de control de televisión.

### Iniciar la música.

- Comando: "Play Music".
- Ejemplo: diga al micrófono el comando "Computer"... "Play Music".

Una vez que se haya activado el modo de música se podrán ejecutar los comandos de control de música.

### **Desactivar micrófono**

Comando: “Disable Speech”

Ejemplo: diga al micrófono el comando “Computer”... “Disable Speech”.

### **Activar micrófono.**

Comando: “Enable Speech”

Ejemplo: diga al micrófono el comando “Computer”... “Enable Speech”.

## **Control de Televisión**

Para separar los comandos de control de TV de los de control general, se utiliza una segunda palabra clave para los comandos que deberá ser usada para indicarle al computador que vamos a ejecutar un comando de TV. Por defecto la palabra clave configurada en el sistema es “TV” y será usada en los ejemplos a continuación.

### **Apagar.**

Comando: “Turn on TV”

Ejemplo: diga al micrófono el comando “Computer”... “Turn on TV”

### **Encender.**

Comando: “Turn off TV”

Ejemplo: diga al micrófono el comando “Computer”... “Turn off TV”

### **Silenciar la televisión.**

Comando: “Mute”

Ejemplo: diga al micrófono el comando “TV”... “Mute”

### **Quitar el silencio.**

Comando: “Unmute”

Ejemplo: diga al micrófono el comando “TV”... “Unmute”

### **Elegir el volumen.**

Comando: “Volume” + *[Valor]*

Ejemplo: diga al micrófono el comando “TV”... “Volume *twenty five*”



### **Cambiar el canal.**

Comando: “Channel” + *[Valor]*

Ejemplo: diga al micrófono el comando “TV”... “Channel *twenty five*”

### **Cambiar la entrada de video.**

El televisor tiene varios métodos de entrada, pero esto varía según el modelo. A continuación se listan las entradas soportadas por el sistema.

- HDMI (del 1 al 4)
- Audio Video (del 1 al 2)
- PC
- TV
- Component (del 1 al 2)

Para controlar cual entrada debe mostrar el televisor, el comando y la sintaxis del mismo se explican a continuación.

Comando: “Show” + *[Valor]*

Ejemplo: diga al micrófono el comando “TV”... “Show *HDMI 1*”

### **Control de Música**

Para indicarle al PC que se va a ejecutar un comando de música, se necesita una tercera palabra clave. La palabra clave por defecto es “Music”, y se la puede cambiar después en el menú de configuración del sistema.

#### **Reproducir música.**

Comando: “Play”

Ejemplo: “Music”... “Play”

#### **Pausar**

Comando: “Pause”

Ejemplo: “Music”... “Pause”

#### **Detener**

Comando: “Stop”

Ejemplo: “Music”... “Stop”

### **Reanudar la reproducción.**

Se debe volver a usar el comando “Play”.

### **Silenciar**

Comando: “Mute”

Ejemplo: “Music”... “Mute”

### **Quitar el silencio**

Comando: “Unmute”

Ejemplo: “Music”... “Unmute”

### **Elegir el volumen.**

Comando: “Volume” + [*Valor*]

Ejemplo: “Music”... “Volume *50*”

### **Búsqueda de artista.**

Comando: “Play Artist” + [*Nombre*]

Ejemplo: “Music”... “Play Artist *Nirvana*”

### **Búsqueda de álbum.**

Comando: “Play Album” + [*Nombre*]

Ejemplo: “Music”... “Play Album *Tourist History*”

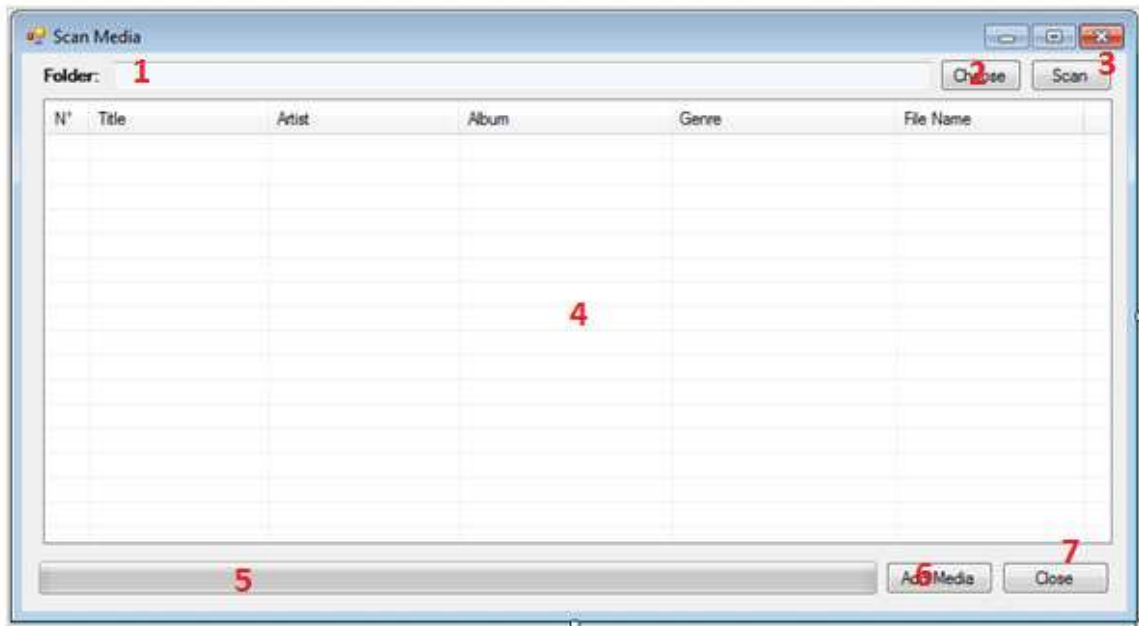
### **Búsqueda de canción.**

Comando: “Play Song” + [*Título Cancion*] + “by” + [*Artista*]

Ejemplo: “Music”... “Play Song *A bad dream* by *Keane*”

## Administración de la Biblioteca Multimedia

Para administrar la biblioteca multimedia del sistema, se debe ingresar al menú de biblioteca multimedia presionando el botón: “Add Media” que está en la pantalla principal del sistema. Ahí se mostrara la pantalla de administración de la colección de canciones del sistema.



Esta pantalla cuenta con los siguientes elementos:

1. Ruta donde se están buscando los archivos multimedia.
2. Botón para escoger el directorio de búsqueda.
3. Botón que inicia la búsqueda de archivos multimedia.
4. Pantalla con los resultados de la búsqueda de los archivos multimedia.
5. Barra que indica el progreso de las operaciones de la pantalla.
6. Botón que graba los archivos seleccionados en el área (4) de la pantalla en la base de datos multimedia del sistema.
7. Botón de salir de este menú.

El proceso para agregar archivos multimedia es simple, se escoge un directorio, se escanea el directorio por archivos compatibles (mp3, mp4, wav, wma). Los archivos se muestran en esta pantalla y se seleccionan cuales se desea agregar a la biblioteca.

Se finaliza con el botón “Add Media” para poder grabar esta información en base de datos. Se repite el proceso hasta tener grabado en la base los archivos que el usuario desea. Y se sale de este menú.

Para eliminar los archivos seleccionados del sistema, se debe dar click derecho en el area (4) de la pantalla, y aparecerá el menú contextual con las opciones:

- Delete Selected Items
- Delete All Folder
- Delete All Database

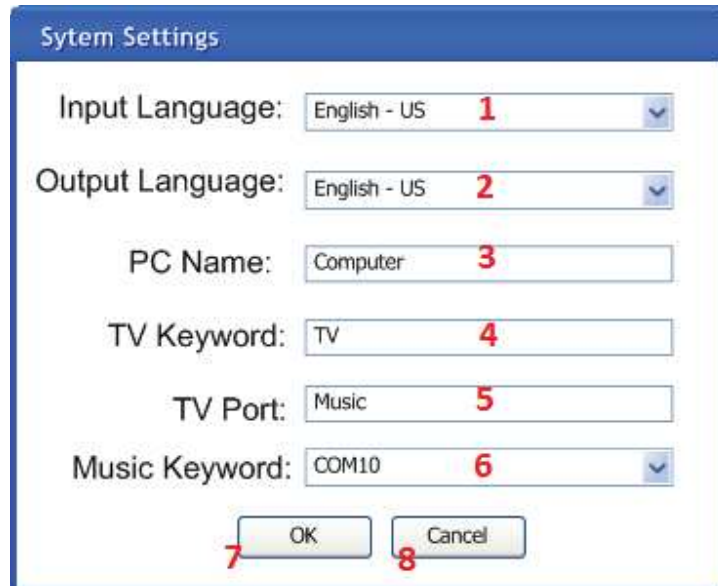
Estas opciones sirven para borrar ítems de la base de datos, la primera elimina los elementos seleccionados del área (4) de la pantalla.

La segunda opción, elimina de la base de datos del sistema todos los ítems contenidos en el directorio de trabajo que se muestra en el área (1) de la pantalla.

La tercera opción limpia totalmente la base de datos multimedia del sistema.

## Configuración avanzada del sistema

Para ingresar a la configuración avanzada del sistema, se debe elegir la opción “Settings” de la pantalla principal del sistema.



Esta pantalla cuenta con los siguientes elementos:

1. Selección de Idioma de Entrada. Es posible activar el reconocimiento de voz en varios idiomas, dependiendo de si el sistema operativo soporta ese idioma.
2. Selección del Idioma de Salida. Aquí se configura en qué idioma se desea que el computador nos “hable”. También si un específico idioma esta soportado o no, dependerá del sistema operativo, pero se pueden instalar voces en idiomas adicionales provistas por terceros, por ejemplo: *Cepstral Voices*.
3. Cuadro de configuración para el nombre del PC.
4. Cuadro de configuración para la palabra clave de los comandos de control de TV.
5. Cuadro de configuración para la palabra clave de los comando de control de música.
6. Selector de puerto COM en donde se ha conectado la TV.
7. Botón de OK. Graba la configuración en el sistema.
8. Botón de Cancelar. Cierra esta pantalla, y no graba ningún cambio a la configuración del sistema.

## Solución a problemas frecuentes

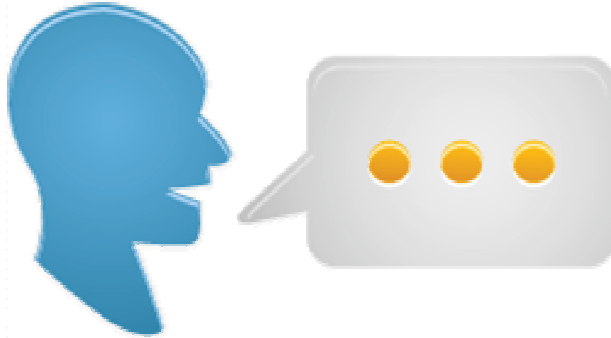
Problema	Causal	Solución
Aparece el dialogo de instalación de base de datos cada vez que inicia la aplicación	Posible desconexión con el software de motor de base de datos	Revise que el motor de base de datos este instalado
		Revise que el servicio de motor de base de datos esté ejecutándose
	Si el motor de base de datos se encuentra en otro equipo, puede ser un problema de red.	Revise que se puede hacer ping al equipo donde se encuentra la base de datos
		Revise que la dirección IP asignada al servidor de base de datos no haya cambiado debido a que se usa un servidor de DHCP en la red.
El sistema no puede escuchar las instrucciones que el usuario habla	Posible error de configuración del micrófono.	Revise que el micrófono se encuentre conectado
		Revise que el micrófono sea el dispositivo de entrada predeterminado para grabación de sonidos en Windows.
		Verifique que el volumen de entrada para el micrófono sea el adecuado, y que no esté el micrófono con la opción "mute" o "silencio".
El sistema reconoce las instrucciones de forma demasiado imprecisa	Baja calidad en la entrada de audio del sistema	Reduzca el ruido ambiental
		Posiciónese correctamente el micrófono al frente de su boca.
		Si el micrófono es de mala calidad, intente con un micrófono de mejor calidad.
El sistema se tarda demasiado tiempo en ejecutar una instrucción dictada por el usuario.	Requisitos de Hardware Limitado	Revise que su equipo cumple con los requisitos mínimos de hardware del sistema
		Alto uso de CPU
		Un programa antivirus está ejecutando una búsqueda completa de virus en el sistema
		La opción de máxima precisión está configurada en el motor de reconocimiento de voz de Windows, revise que este en un punto medio entre rendimiento y precisión.

Problema	Causal	Solución
El sistema rechaza las instrucciones que habla el usuario.	Error de configuración del idioma del sistema.	Revisar que el motor de reconocimiento de voz de Windows coincida con el idioma en el cual desea hablar el usuario, por ejemplo, en un Windows Vista en español, no se puede ejecutar reconocimiento de voz en alemán y viceversa.
No se puede controlar la TV	Error de conexión con la TV	El cable de conexión con TV esta desconectado.
		La TV no es compatible con el set de comandos de la serie LG 30R.
		El puerto COM configurado en el sistema no coincide con el puerto COM al cual el TV está conectado.

## **Anexo 4.**



# Especificación de Requerimientos de Software



## Speech Media Controller

Sistema de Control por Comandos de Voz de las Funciones  
Multimedia del Hogar.

Autor:

José Adrián Salazar

Septiembre 2011

## Historial de Revisiones

Versión	Fecha	Nombres	Comentarios
1.0	2011-09-17	Adrián Salazar	Creación del Documento

## Tabla de Contenidos

1.	Introducción.....	4
1.1.	Descripción del Producto.....	4
2.	Requerimientos Específicos.....	4
2.1.	Interfaces Externas.....	4
2.1.1.	Interfaz de Usuario.....	4
2.1.2.	Interfaz de Hardware.....	4
2.1.3.	Interfaz de Software.....	5
2.1.4.	Protocolos de Comunicación.....	5
2.2.	Características del Producto de Software.....	5
2.2.1.	Módulo de Reconocimiento de Voz.....	5
2.2.2.	Módulo de Control de Funciones Multimedia.....	5
2.2.3.	Módulo de Control de Televisor.....	6
2.2.4.	Módulo de Configuración y Administración.....	6
2.3.	Atributos del Sistema de Software.....	6
2.3.1.	Confiabilidad.....	6
2.3.2.	Disponibilidad.....	6
2.3.3.	Seguridad.....	6
2.3.4.	Requerimientos de Mantenimiento.....	7
2.3.5.	Portabilidad.....	7
2.3.6.	Rendimiento.....	7
2.4.	Requerimientos de Base de Datos.....	7
3.	Material Adicional.....	7

## 1. Introducción

El siguiente documento tiene como fin, el especificar los requerimientos de un producto de software destinado a controlar las funciones multimedia de audio y video en el hogar, usando reconocimiento de voz.

El software está destinado al usuario de casa, el mismo que podrá a través de un set de comandos de voz, controlar su reproductor de música Windows Media, y su televisor LCD o Plasma de la marca LG, todo esto apalancado en el software que se ejecuta en su computador de hogar.

### 1.1. Descripción del Producto

El producto de software comprende 3 grandes sub sistemas:

- Sistema de Reconocimiento de Voz
- Sistema reproductor Multimedia
- Sistema de control del Hardware para televisores LG.

La principal tarea de este aplicativo será el de orquestar a los sub sistemas y de transmitir las ordenes que provienen desde el usuario hasta el sub sistema a cargo de realizar la tarea deseada.

El usuario dispondrá de un set de instrucciones pre-definidas, con las cuales, podrá ejecutar comandos a través de su voz, los cuales serán interpretados por la aplicación.

## 2. Requerimientos Específicos

### 2.1. Interfaces Externas

#### 2.1.1. Interfaz de Usuario

La interfaz de usuario será una interfaz híbrida. Interfaz Gráfica e Interfaz de Voz.

La interfaz gráfica será desarrollada a manera de aplicación de escritorio de Windows. En esta aplicación, el usuario realizara las tareas de configuración y de ingreso de parámetros de la aplicación.

La segunda interfaz, será una interfaz de voz, en la cual el usuario podrá navegar a través de un menú de opciones únicamente a través del habla. Esta será la principal interfaz del sistema.

#### 2.1.2. Interfaz de Hardware

El sistema deberá interactuar con el hardware de los televisores Plasma y LCD del fabricante LG Electronics.

Esta línea de productos se conecta al computador a través de un puerto RS-232, mejor conocido como puerto COM. La aplicación debe ser capaz de enviar y recibir comandos a través de este puerto.

#### 2.1.3. Interfaz de Software

El sistema, hará uso de dos *API's* de programación:

El *Speech API* le permitirá al sistema, procesar la voz del usuario y traducirla a comandos.

El Windows Media Player SDK le permite a la aplicación el control de las funciones multimedia de este reproductor de una manera programable.

#### 2.1.4. Protocolos de Comunicación

El sistema deberá tener la capacidad de enviar, recibir e interpretar los mensajes descritos en el apéndice B del manual de operación de los Televisores LG, set de comandos para RS232C.

### 2.2. Características del Producto de Software

El sistema deberá estar dividido en 4 módulos, los cuales se listan a continuación.

#### 2.2.1. Módulo de Reconocimiento de Voz

El sistema deberá recibir el ingreso de voz desde el micrófono del sistema y ejecutar el reconocimiento de un set de instrucciones limitadas.

El sistema no requerirá entrenamiento previo del usuario.

El sistema deberá separar cada set de instrucciones en tres categorías: General, TV y Música.

El sistema deberá reconocer a cada uno de este conjunto de comandos y consecuentemente deberá realizar una acción predeterminada.

Si el sistema no llegara a entender un comando, deberá emitir una retroalimentación al usuario diciéndole que no se ha podido interpretar el comando.

Si el sistema no puede ejecutar un comando, deberá comunicarle también al usuario que su comando no se ha podido ejecutar.

El sistema deberá reconocer comandos en ingles, pero no deberá estar limitado a este lenguaje.

#### 2.2.2. Módulo de Control de Funciones Multimedia

El sistema deberá poder controlar las funciones del reproductor de Windows media instalado en el computador.

Las funciones a controlar serán: Reproducción, Pausa, Detener, Reanudar y control de volumen.

El sistema podrá realizar búsquedas por artista, por álbum, y por canción.

#### 2.2.3. Módulo de Control de Televisor

El sistema deberá poder controlar televisores de la marca LG de la serie 30R.

Las funciones a controlar será las siguientes: Encendido, apagado, control de la fuente de ingreso, sintonización de canal, control del volumen.

#### 2.2.4. Módulo de Configuración y Administración

El sistema deberá contar con un modulo de configuración de los parámetros generales del sistema.

Los parámetros que se pueden configurar serán los siguientes: Puerto de conexión de la televisión, idioma en el cual se realiza el reconocimiento de voz, idioma en el cual el sistema genera la retroalimentación al usuario.

### 2.3. Atributos del Sistema de Software

#### 2.3.1. Confiabilidad

El sistema deberá reconocer un promedio de 80% de los comandos de voz pronunciados por el usuario. Se asume que el entorno es un entorno favorable para la captura de audio desde el micrófono esto es, estando el micrófono correctamente posicionado, a una distancia adecuada del interlocutor, y con ruido de fondo moderado.

#### 2.3.2. Disponibilidad

El sistema debe estar disponible al usuario de manera automática al activarse el equipo de cómputo del hogar, y habiéndose ya finalizado la carga del sistema operativo y de haber iniciado correctamente sesión en su sistema.

Un comando mal interpretado o un error de comunicación con el hardware o con las interfaces de software no deberán incurrir en caídas del sistema. Ante un evento de esta naturaleza, el sistema deberá notificar al usuario vía audio que esta(s) función(es) no se encuentran disponibles.

#### 2.3.3. Seguridad

La interfaz de voz del sistema deberá estar abierta a cualquier usuario del hogar, debido a la naturaleza compartida del ambiente y del espacio físico donde se encuentra, por ejemplo: sala de estar. El propósito del sistema es hacer la experiencia multimedia del hogar más fácil; implementar controles de seguridad para su uso, convertirían este sistema en algo poco amigable.

#### 2.3.4.Requerimientos de Mantenimiento

El sistema no debe requerir mantenimiento frecuente. Su instalación y actualización deben ser de fácil ejecución ya que el usuario final será el que realice esta tarea.

#### 2.3.5.Portabilidad

El sistema deberá ser compatible con las siguientes versiones de Microsoft Windows:

- Windows XP Home 32 bit.
- Windows XP Professional 32bit.
- Windows Vista Business/Ultimate 32bit.
- Windows 7 Home Premium/Professional/Ultimate 32bit.

Sin embargo no se restringe la posibilidad de que el sistema sea portado a futuras versiones de Microsoft Windows o a las versiones mencionadas de Windows anteriormente, a 64bit.

#### 2.3.6.Rendimiento

El sistema debe capturar, interpretar, reconocer y ejecutar el comando hablado por el usuario en un tiempo promedio no mayor a los 3 segundos.

Este tiempo se lo considera aceptable y para su medición se asume que el equipo de cómputo está bajo los siguientes parámetros de operación recomendados:

- Una potencia de cómputo de 2.0 GHz mínimo.
- Cantidad de memoria RAM disponible de al menos 1.5 GB.
- Uso de procesador, sin el sistema en ejecución de máximo del 10%.
- Ningún otro sistema de reconocimiento de voz se encuentra ejecutándose en paralelo.
- Ningún proceso de desfragmentación de disco, de escaneo de virus, de uso intensivo de disco duro o de procesador o de tarjeta gráfica se ejecuta en paralelo.
- El equipo no estará virtualizado. No se garantiza el optimo rendimiento del software en maquinas virtuales.

Cualquier medición que haya sido tomada en un equipo por debajo de estos parámetros de operación deberá ser descartada.

#### 2.4. Requerimientos de Base de Datos

El sistema deberá soportar la versión de SQL Server 2008 Express o Superior.

### 3. Material Adicional

Basarse en la implementación de comunicación con televisores LG en el manual de usuario de la serie LG30R.

## **Anexo 5.**



# APÉNDICE

## CÓDIGOS DE IR

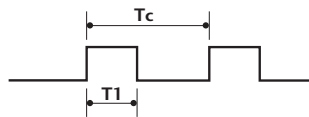
### 1. Cómo conectar

- Conecte su control remoto de cable al puerto de control remoto en la TV.

### 2. Códigos de IR de Control Remoto

#### ■ Salida de la forma de onda

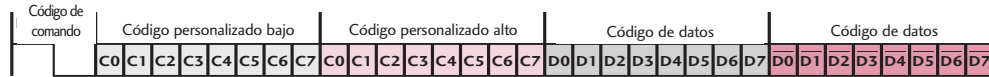
Pulso simple, modulado con la señal de 37,917 kHz en 455 kHz



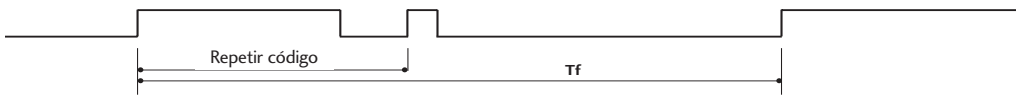
Frecuencia de transportador  
 $FCAR = 1/TC = fOSC/12$   
 Proporción debida =  $T1/TC = 1/3$

#### ■ Configuración de cuadro

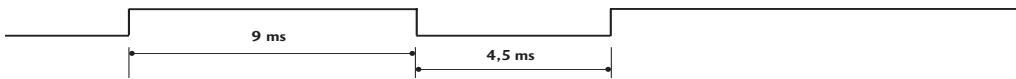
1r cuadro



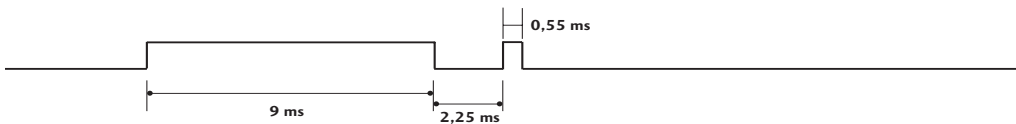
Cuadro repetido



#### ■ Código de líder



#### ■ Código de repetición

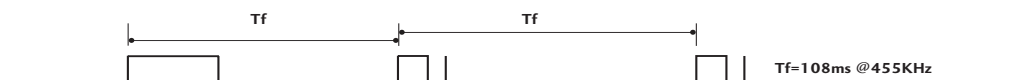


#### ■ Descripción de bits



#### ■ Intervalo de cuadros: Tf

La forma de onda es transmitida mientras una tecla está presionado.



Código (Hexa)	Función	Nota
08	POWER	Botón control Remoto (Encendido/apagado)
45	Q.MENU	Botón control Remoto
43	MENU	Botón control Remoto
0B	INPUT	Botón control Remoto
10-19	Teclas de números 0-9	Botón control Remoto
09	MUTE	Botón control Remoto
02	VOL +	Botón control Remoto
03	VOL -	Botón control Remoto
00	CH ^	Botón control Remoto
01	CH v	Botón control Remoto
1E	FAV	Botón control Remoto
40	^	Botón control Remoto
41	v	Botón control Remoto
07	<	Botón control Remoto
06	>	Botón control Remoto
7C	X STUDIO	Botón control Remoto
44	ENTER	Botón control Remoto
28	BACK	Botón control Remoto
7E	SIMPLINK	Botón control Remoto
30	AV MODE	Botón control Remoto
79	RATIO	Botón control Remoto
95	ENERGY SAVING	Botón control Remoto
53	LIST	Botón control Remoto
1A	Q.VIEW	Botón control Remoto
BA	FREEZE	Botón control Remoto
72	ROJO	Botón control Remoto
63	AMARILLO(PLAY)	Botón control Remoto
61	AZUL(PAUSE)	Botón control Remoto
71	VERDE (STOP)	Botón control Remoto
8F	Rew ◀◀	Botón control Remoto
8E	FF ▶▶	Botón control Remoto
5E	CLEAR VOICE II	Botón control Remoto
B1	Stop ■	Botón control Remoto
B0	Play ▶	Botón control Remoto
0F	TV	Botón control Remoto
5B	EXIT	Botón control Remoto

Excepto 42/50PJ250R

Para 42/50PJ250R

## APÉNDICE

### CONFIGURACIÓN DE DISPOSITIVO EXTERNO DEL CONTROL

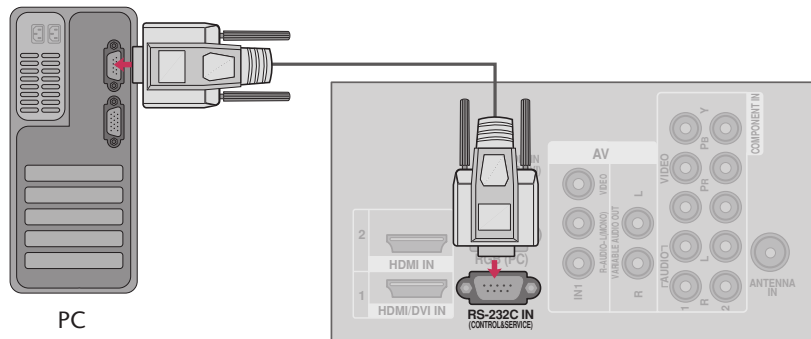
El puerto RS-232 permite conectar el conector de entrada RS-232C a un dispositivo de control externo (como una computadora o un sistema de control de A/V) para controlar en forma externa las funciones del televisor.

- Nota: El RS-232 de esta unidad está diseñado para ser utilizado con hardware y software de control de RS-232 de terceros.

Las instrucciones que aparecen a continuación se entregan para servir de ayuda con el software de programación o para probar la funcionalidad del uso de software telenet.

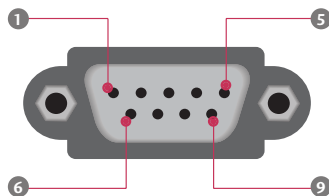
- En modo USB y Bluetooth, no es compatible con función RS-232C.
- La imagen puede ser diferente a la de su televisor.

#### Instalación de RS-232C



#### Tipo del Conector; Macho D-Sub 9-Pines

No.	Nombre de Pin
1	No conexión
2	RXD (Recepción de datos)
3	TXD (Transmisión de datos)
4	DTR (Lado DTE listo)
5	GND
6	DSR (Lado DCE listo)
7	RTS (Listo para enviar)
8	CTS (Aprobado para enviar)
9	No Conexión

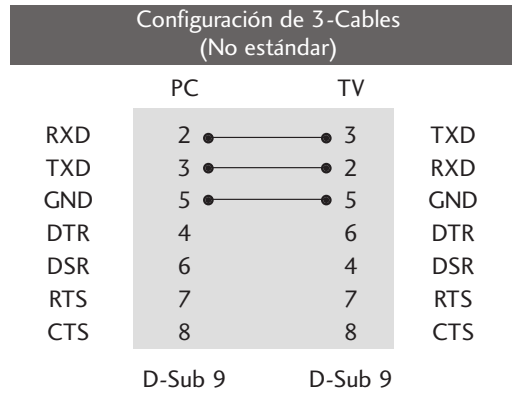
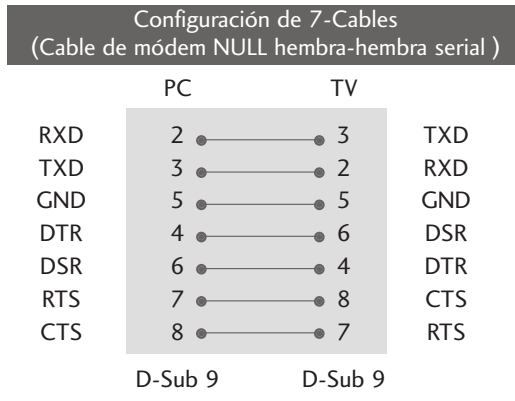


#### Comunicación de los Parámetros

- Velocidad de banda : 9 600 b/s (UART)
- Longitud de datos : 8 bits
- Paridad : No se aplica
- Bit de detención : 1 bit
- Código de comunicación : Código ASCII
- Use un cable cruzado (reverso).

## Configuraciones de RS-232C

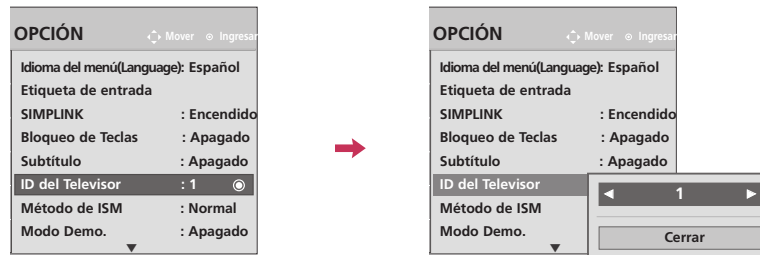
Se puede utilizar cualquiera de los cables a continuación.



## Configuración de ID

Use esta función para especificar un número de ID de la unidad.

Refiérase a "Mapeo de datos reales".



- 1 Seleccione **OPCIÓN**.
- 2 Seleccione **ID del Televisor**.
- 3 Seleccione el número deseado de ID del monitor. Rango de ajuste de **ID del Televisor 1 - 99**.
- 4 Regresar al menú anterior.
- Regresar a la pantalla del televisor.

**\* Mapeo de datos reales 1**

00	: Paso 0
⋮	
A	: Paso 10
⋮	
F	: Paso 15
10	: Paso 16
⋮	
64	: Paso 100
⋮	
6E	: Paso 110
⋮	
73	: Paso 115
74	: Paso 116
⋮	
C7	: Paso 199
⋮	
FE	: Paso 254
FF	: Paso 255

**\* Mapeo de datos reales 2**

00	: -30
01	: -29
02	: -28
⋮	
E	: 0
⋮	
28	: +10
⋮	
3A	: +28
3B	: +29
3C	: +30
⋮	
4E	: N/A
4F	: N/A
50	: N/A

# APÉNDICE

## Lista de Referencia de los Comandos

	COMANDO 1	COMANDO 2	DATO (Hexadecimal)		COMANDO 1	COMANDO 2	DATO (Hexadecimal)
01. Alimentación	k	a	00 ~ 01	14. Temperatura de color	x	u	00 ~ 64
02. Relación de Aspecto	k	c	Refiérase a p.113	15. Agudos	k	r	00 ~ 64
03. Modo Pantalla	k	d	00 ~ 01, 10	16. Graves	k	s	00 ~ 64
04. Silenciador de volumen	k	e	00 ~ 01	17. Estado anormal	k	z	FF
05. Control de volumen	k	f	00 ~ 64	18. Método de ISM	j	p	Refer to p.115
06. Contraste	k	g	00 ~ 64	19. Auto-configuración	j	u	01
07. Brillo	k	h	00 ~ 64	20. Bajo consumo (Ahorro de energía)	j	q	00 ~ 04
08. Color	k	i	00 ~ 64	21. Selección de canal	m	a	Refer to p.116
09. Tinte	k	j	00 ~ 64	22. Agr./Eli. de canal	m	b	00 ~ 01
10. Nitidez	k	k	00 ~ 64	23. Botones	m	c	Refer to p.116
11. Selección de OSD	k	l	00 ~ 01	24. Selección de entrada	x	b	Refer to p.117
12. Modo bloqueado de control remoto	k	m	00 - 01				
13. Balance	k	t	00 ~ 64				

## Protocolo de Transmisión /Recepción

### Transmisión

**[Command1][Command2][ ][Set ID][ ][Data][Cr]**

[Comando 1] : Primer comando para controlar la TV.(j,k,m o x)

[Comando 2] : Segundo comando para controlar la TV.

[ID del Televisor]: Ud. puede ajustar la ID de unidad para seleccionar el número deseado de ID de la TV en el menú Especial. Rango de ajuste es de 1 a 99. Cuando seleccione la ID de unidad '0', cada TV conectada será controlada. ID de unidad está indicado en decimal (1 - 99) en el menú en Hexadecimal (0x0 - 0x63) en la transmisión/recibiendo el protocolo.

[DATO] : Para transmitir los datos de comando.

Transmite el dato 'FF' a leer estados de comando

[Cr] : Retorno

Código ASCII como '0x0D'

[ ] : Código ASCII como 'espacio (0x20)'

### Reconocimiento de OK

**[Command2][ ][Set ID][ ][OK/NG][Data][x]**

La TV transmite ACK (reconocimiento) basando en este formato cuando recibe un dato normal. En este momento, si el dato está en el modo de leer, esto indica el estado de dato actual. Si el dato está en el modo de escribir, esto regresa los datos de la computadora.

### Reconocimiento de NG

**[Command2][ ][Set ID][ ][NG][Data][x]**

La TV transmite ACK (reconocimiento) basando en este formato cuando recibe un dato anormal de funciones no-viable o errores de comunicación.

[Comando 2] : Use como comando.

[ID del Televisor] : Use el carácter pequeño, si la ID definida es 10, enviará '0'; 'a'

[DATO] : Use el carácter pequeño, si los datos son 0xab, enviará 'a'; 'b'.

[NG] : Use el carácter grande

**01. Alimentación (Comando: k a)**

Para controlar la potencia Encendida/Apagada de la TV.

*Transmisión* [k][a][ ][ID del Televisor][ ][Dato][Cr]

Dato 00 : Apagado      Dato 01 : Encendido

*Reconocimiento* [a][ ][ID del Televisor][ ][OK/NG][Dato][x]

Para mostrar la potencia Encendida/Apagada.

*Transmisión* [k][a][ ][ID del Televisor][ ][FF][Cr]

*Reconocimiento* [a][ ][ID del Televisor][ ][OK/NG][Dato][x]

\* Si existe un período de retraso entre el comando y el mensaje de confirmación durante la secuencia de encendido, este debería ser descrito como "Nota: en este modelo, la TV enviará la confirmación tras completarse el proceso de encendido. Podría existir un período de retraso entre el comando y la confirmación."

**02. Proporción de aspecto (Comando: k c)**

Para ajustar el formato de pantalla. (Formato de la imagen principal) Ud.

Puede hacer la Relación de Aspecto con el menú Imagen.

*Transmisión* [k][c][ ][ID del Televisor][ ][Dato][Cr]

Dato 01 : 4:3                      04: Zoom  
 02: 16:9                      10: Zoom cine 1  
 09: Sólo búsqueda            1F: Zoom cine 16

*Reconocimiento* [c][ ][ID del Televisor][ ][OK/NG][Dato][x]

\* Al utilizar la entrada RGB, debe seleccionar entre el formato de imagen en pantalla 16:9 o 4:3.

\* Esta función sólo se opera en los modos de Componente/HDMI(720progresivo/1 080entrelazado/1 080progresivo) input source.

**03. Modo de pantalla (Comando: k d)**

Para seleccionar mudo de pantalla encendido/apagado.

*Transmisión* [k][d][ ][ID del Televisor][ ][Dato][Cr]

Dato 00 : Mudo encendido de pantalla(Imagen apagada)

Dato 01 : Mudo apagado de pantalla(Imagen encendida)

Dato 00 : Silencio de video off (desc.)

Dato 10 : Silencio de video on (act.)

*Reconocimiento* [d][ ][ID del Televisor][ ][OK/NG][Dato][x]

\* Sólo en el caso de que la función Silencio de video esté activada, el televisor mostrará la visualización en pantalla (OSD) Pero en el caso de que la función Silencio de video esté desactivada, el televisor no mostrará la visualización en pantalla (OSD).

**04. Silenciador de volumen (Comando: k e)**

Para controlar el encendido/apagado del silenciador de volumen, También puede ajustar el silenciador utilizando el botón MUTE en el control remoto.

*Transmisión* [k][e][ ][ID del Televisor][ ][Dato][Cr]

Dato 00: Encendido del silenciador de volumen (Sin volumen)

Dato 01: Apagado del silenciador de volumen (Con volumen)

*Reconocimiento* [e][ ][ID del Televisor][ ][OK/NG][Dato][x]

**05. Control de volumen (Comando: k f)**

Para ajustar el volumen.

Ud. también puede ajustar el volumen con los botones de volumen del control remoto.

*Transmisión* [k][f][ ][ID del Televisor][ ][Dato][Cr]

Dato Min : 00 - Máx : 64

Refiérase a 'Mapeo de datos reales1'.

*Reconocimiento* [f][ ][ID del Televisor][ ][OK/NG][Dato][x]

**06. Contraste (Comando: k g)**

Para ajustar el contraste de la pantalla.

También puede ajustar el contraste en el menú Imagen.

*Transmisión* [k][g][ ][ID del Televisor][ ][Dato][Cr]

Dato Min : 00 - Máx : 64

Refiérase a 'Mapeo de datos reales1'.

*Reconocimiento* [g][ ][ID del Televisor][ ][OK][Dato][x]

**07. Brillo (Comando: k h)**

Para ajustar el brillo de la pantalla.

También puede ajustar el brillo en el menú Imagen.

*Transmisión* [k][h][ ][ID del Televisor][ ][Dato][Cr]

Dato Min : 00 - Máx : 64

Refiérase a 'Mapeo de datos reales1'.

*Reconocimiento* [h][ ][ID del Televisor][ ][OK/NG][Dato][x]

**08. Color (Comando: k i)**

Para ajustar el color de pantalla.

También puede ajustar el brillo en el menú Imagen.

*Transmisión* [k][i][ ][ID del Televisor][ ][Dato][Cr]

Dato Min : 00 - Máx : 64

Refiérase a 'Mapeo de datos reales1'.

*Reconocimiento* [i][ ][ID del Televisor][ ][OK/NG][Dato][x]

**09. Tinte (Comando: k j)**

Para ajustar el tinte de la pantalla.

Ud. puede ajustar el tinte en el menú Imagen.

*Transmisión* [k][j][ ][ID del Televisor][ ][Dato][Cr]

Dato Rojo : 00 - Verde : 64

Refiérase a 'Mapeo de datos reales1'.

*Reconocimiento* [j][ ][ID del Televisor][ ][OK/NG][Dato][x]

# APÉNDICE

## 10. Nitidez (Comando: k k)

Para ajustar la nitidez de la pantalla.

También puede ajustar la nitidez en el menú Imagen.

*Transmisión* [k][k][ ][ID del Televisor][ ][Dato][Cr]

Dato Min: 00 - Máx: 64

Refiérase a 'Mapeo de datos reales1'.

*Reconocimiento* [k][ ][ID del Televisor][ ][OK/NG][Dato][x]

## 11. Selección de OSD (Comando: k l)

Para seleccionar OSD (On Screen Display)

encendido/apagado controlando remotamente.

*Transmisión* [k][l][ ][ID del Televisor][ ][Dato][Cr]

Dato 00: OSD apagado

Dato 01: OSD encendido

*Reconocimiento* [l][ ][ID del Televisor][ ][OK][Dato][x]

## 12. Modo bloqueado de control remoto (Comando: k m)

Para bloquear el control remoto y los controles del panel frontal en la unidad. (Si ud. no está usando el control remoto y tampoco los controles del panel frontal en la unidad, use este modo. Cuando la potencia principal enciende y apaga, el bloqueo de control remoto será liberado.)

*Transmisión* [k][m][ ][ID del Televisor][ ][Dato][Cr]

Dato 00: Bloquear Apagado

Dato 01: Bloquear Encendido

*Reconocimiento* [m][ ][ID del Televisor][ ][OK/NG][Dato][x]

## 13. Balance (Comando: k t)

Para ajustar el balance.

También puede ajustarse el balance en el menú audio.

*Transmisión* [k][t][ ][ID del Televisor][ ][Dato][Cr]

Dato Min: 00 - Máx: 64

Refiérase a 'Mapeo de datos reales1'.

*Reconocimiento* [t][ ][ID del Televisor][ ][OK/NG][Dato][x]

## 14. Temperatura de color (Comando: x u)

Para ajustar la temperatura de color.

También puede ajustar la temperatura de color en el Menú Imagen.

*Transmisión* [x][u][ ][ID del Televisor][ ][Dato][Cr]

Dato W50 : 00 ~ C50 : 64

*Reconocimiento* [u][ ][ID del Televisor][ ][OK/NG][Dato][x]

## 15. Agudos (Comando: k r)

Para ajustar los agudos.

También puede ajustar los agudos en el menú audio.

*Transmisión* [k][r][ ][ID del Televisor][ ][Dato][Cr]

Dato Min: 00 - Máx: 64

Refiérase a 'Mapeo de datos reales1'.

*Reconocimiento* [r][ ][ID del Televisor][ ][OK/NG][Dato][x]

## 16. Graves (Comando: k s)

Para ajustar los graves.

También puede ajustar los graves en el menú audio.

*Transmisión* [k][s][ ][ID del Televisor][ ][Dato][Cr]

Dato Min: 00 - Máx: 64

Refiérase a 'Mapeo de datos reales1'.

*Reconocimiento* [s][ ][ID del Televisor][ ][OK/NG][Dato][x]

## 17. Estado anormal (Comando: k z)

Para reconocer un estado anormal.

*Transmisión* [k][z][ ][ID del Televisor][ ][FF][ ][Cr]

Dato FF: Leer

*Reconocimiento* [z][ ][ID del Televisor][ ][OK/NG][Dato][x]

Dato 00: Normal (Encendido y existe la señal)

Dato 01: No señal (Encendido)

Dato 02: Apague la TV con el control remoto

Dato 03: Apague la TV con la función de temporizador para dormir

Dato 04: Apague la TV con la función RS-232C

Dato 05: 5V bajo

Dato 06: AC bajo

Dato 07: Apague la TV con la función de alarma de abanico (opción)

Dato 08: Apague la TV con la función de tiempo apagado

Dato 09: Apague la TV con la función de auto-dormido

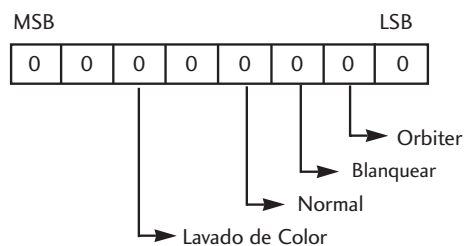
Dato a: Apague la TV para detectar el AV.

\* Esta función es para "leer sólo".

## 18. Método de ISM (Comando: j p)

Para evitar tener una imagen fija permanente en pantalla.

*Transmisión* [j][p][ ][ID del Televisor][ ][Dato][Cr]



7	6	5	4	3	2	1	0	Función
				0	0	1	0	Orbiter
				0	1	0	0	Blanquear
				1	0	0	0	Normal
		1	0	0	0	0	0	Lavado de Color

Dato 02: Orbiter

Dato 04: Blanquear

Dato 08: Normal

Dato 20: Lavado de Color

\* No pueden ajustarse más de dos tipos de funciones al mismo tiempo.

*Reconocimiento* [p][ ][ID del Televisor][ ][OK/NG][Dato][x]

## 19. Auto-configuración (Comando: j u) -Except 42/50P250R

Para ajustar la posición de imagen y minimizar el temblor de imagen automáticamente. Sólo se trabaja en el modo RGB.

*Transmisión* [j][u][ ][ID del Televisor][ ][Dato][Cr]

Dato 01: Para ajustar

*Reconocimiento* [u][ ][ID del Televisor][ ][OK/NG][Dato][x]

## 20. Ahorro de Energía (Comando: j q)

Para reducir el consumo de potencia del monitor.

*Transmisión* [j][q][ ][ID del Televisor][ ][Dato][ ][Cr]

Dato 00: Ahorro de energía Apagado

01: Ahorro de energía: Mínimo

02: Ahorro de energía: Medio

03: Ahorro de energía: Máximo

05: Desconexión de pantalla

10: Sensor Inteligente

*Reconocimiento* [q][ ][ID del Televisor][ ][OK/NG][Dato][x]

### \* Mapeo de datos reales 1

00 : Paso 0

⋮

A : Paso 10

⋮

F : Paso 15

10 : Paso 16

⋮

64 : Paso 100

⋮

6E : Paso 110

⋮

73 : Paso 115

74 : Paso 116

⋮

C7 : Paso 199

⋮

FE : Paso 254

FF : Paso 255

### \* Mapeo de datos reales 2

00: -30

01: -29

02: -28

1E: 0

⋮

28: +10

3A: +28

3B: +29

3C: +30

4E: N/A

4F: N/A

50: N/A



# APÉNDICE

## 21 . Selección de canal (Comando: m a)

Sintoniza canales en los números físicos siguientes.

*Transmisión* [m][a][ ][ID del Televisor][ ][Dato0][ ][Dato1][ ][Dato2][ ][Dato3][ ][Dato4][ ][Dato5][Cr]

Dato0: Número de canal físico(\*transmitir por código hexadecimal)

NTSC air: 02-45

NTSC cable: 01, 0e-7d

Dato1, Dato2: Mayor número de canal

Dato1: Bytes altos

Dato2: Bytes bajos

Número de canal en dos partes: Número mayor - Número menor

Número de canal en una parte:

En caso de utilizar un número de canal en una parte, el canal menor no será necesario.

Dato3 & 4: Número de canal menor

No se necesita para NTSC.

Dato3: Bytes altos

Dato4: Bytes bajos

Dato5:

Atributo

7	Imagen principal/secundaria	6	Canal de dos/una parte	5	Uso de canal físico	4 Reservado	3	2	1	0	7
0	Principal	0	Dos	0	Uso	X	0	0	0	0	NTSC air
1	Secundaria	1	Uno	1	Sin uso	X	0	0	0	1	NTSC cable

Ejemplo)

- Comando: ma 00 23 xx xx xx xx 01 atributo (0x01): datos de imagen principal, dos partes (no es obligatorio), uso de canal físico, cable NTSC 'xx': en caso de sintonización de canal analógico el número de canal mayor o menor no importará

*Reconocimiento* [a][ ][ID del Televisor][ ][OK][Dato0][Dato1][Dato2][Dato3][Dato4][Dato5][x][a][ ][ID del Televisor][ ][NG][Dato0][x]

## 22. Agr./Eli. de canal (Comando: m b)

Para agregar y eliminar canales.

*Transmisión* [m][b][ ][ID del Televisor][ ][Dato][Cr]

Dato00 : Borrar programa      01 : Memorizar programa

*Reconocimiento* [b][ ][ID del Televisor][ ][OK/NG][Dato][x]

## 23. Botones (Comando: m c)

Para mandar el código de botón remotó IR.

*Transmisión* [m][c][ ][ID del Televisor][ ][Dato][Cr]

Dato: Código de botones - Refiérase a la página 103.

*Reconocimiento* [c][ ][ID del Televisor][ ][OK/NG][Dato][x]

## 24. Selección de entrada (Comando: x b)

### (Entrada de la imagen principal)

Para seleccionar la fuente de entrada para la TV.

*Transmisión* [x][b][ ][ID del Televisor][ ][Dato][Cr]

Dato: Estructura

MSB				LSB			
0	0	0	0	0	0	0	0

└─ Entrada externa ─┘

└─ No. de entrada ─┘

Entrada externa				Dato
0	0	0	1	Analog
0	0	1	0	AV
0	1	0	0	Componente
0	1	1	0	RGB-PC
1	0	0	1	HDMI

Excepto 42/50P]250R

Número de entrada				Dato
0	0	0	0	Input1
0	0	0	1	Input2
0	0	1	0	Input3
0	0	1	1	Input4

*Reconocimiento* [b][ ][ID del Televisor][ ][OK/NG][Dato][x]