

Universidad de las Ciencias Informáticas

Facultad 2



Componente para realizar el dictado automático en la  
elaboración de los informes diagnóstico.

Trabajo de Diploma para optar por el Título de  
Ingeniero en Ciencias Informáticas

Autores: Damisel Pozo Pozo.  
Yasmany Reyes Gonzáles.

Tutores: Ing. Yuniesky Armentero Moreno.  
Ing. Dismey Saavedra López.

La Habana, 20 de Junio de 2014  
“Año 56 de la Revolución”

## PENSAMIENTO



*"...lo fundamental es hacer algo nuevo cada día y luego perfeccionar lo que se ha hecho el día anterior..."*

Ernesto Guevara de la Serna

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 20 días del mes de Junio del año 2014.

Autores:

---

Damisel Pozo Pozo

---

Yasmany Reyes Gonzáles

Tutores:

---

Ing. Yuniesky Armentero Moreno

---

Ing. Dismey Saavedra López

## DATOS DE CONTACTO

**Ing. Yuniesky Armentero Moreno:** Ingeniero en Ciencias Informáticas, graduado en la Universidad de las Ciencias Informáticas en el año 2007. Posee la categoría docente de Profesor Asistente. Actualmente es profesor de PID III, ha impartido otras asignaturas como son, Introducción a la Programación, Programación I, Gráfico por Computadoras, DHC, Programación 4 y además varios cursos optativos y del segundo perfil en la Facultad 6 y 7. Se desempeñó como coordinador de PP4-5 en el Departamento de Producción de Software Médico Imagenológico (SWMI), Centro de Informática Médica (CESIM) y actualmente es jefe de asignatura.

Correo electrónico: yarmentero@uci.cu

**Ing. Dismey Saavedra López:** Ingeniero en Ciencias Informáticas, graduado en la Universidad de las Ciencias Informáticas en el año 2008. Posee la categoría docente de Profesor Instructor. Actualmente es profesora de Álgebra Lineal en la facultad 7, ha impartido otras asignaturas como son, Matemática Discreta 1. Se desempeñó como jefa de asignatura de Álgebra Lineal en el Departamento de Ciencias Básicas y actualmente es profesora de Álgebra Lineal de 3 grupos y secretaria general del comité de base de Ciencias Básicas. Se encuentra en la fase final de su tesis de maestría.

Correo electrónico: dsaavedra@uci.cu

## DEDICATORIA

*Le dedico este trabajo a mis amores:*

*A mi madre, por hacerme la persona que soy, por su dedicación, te idolatro.*

*A mi hermano, por estar siempre en mi corazón y en mi vida.*

*De Damisel.*

*A mis padres, por haberme apoyado y guiarme siempre por el buen camino.*

*A mi familia y a mi novia, por su comprensión y dedicación.*

*De Yasmany.*

## AGRADECIMIENTOS

*A mi madre, por ser mi todo, te amo.*

*A mi hermano, por ser mi ángel de la guardia.*

*A mimá y a papi, por ser su niña preferida.*

*A tía Ana y Mercedes, por tenerme como una hija más.*

*A mi niño entero y al resto de la familia.*

*A Osmany, por su apoyo incondicional desde que llegó a mi vida.*

*A todos mis amigos, por compartir sus días conmigo.*

*A Yovanotti, por su ayuda en este trabajo.*

*Y a esta revolución.*

*De Damisel.*

*A mis padres, por haberme apoyado y guiarme siempre por el buen camino.*

*A mi familia y a mi novia, por su comprensión y dedicación.*

*De Yasmany.*

## RESUMEN

Los Sistemas de Información Radiológica (RIS, por sus siglas en inglés) tienen entre sus funciones principales la de proporcionar un informe diagnóstico sobre el estudio que se realiza a un paciente. El Centro de Informática Médica (CESIM) posee como una de sus aplicaciones desarrolladas, un RIS, el cual utiliza un subsistema Reportador para la gestión de estos informes.

En la actualidad existen dos flujos de trabajo básicos para la realización de informes diagnóstico, por transcripción y sin transcripción. En ambos casos, se perciben problemas que pueden afectar la terminación correcta y la entrega de los mismos, como son la ralentización y la mala calidad debido a errores en su confección.

Por la existencia de estos problemas, se decide desarrollar un componente que permita agilizar el dictado automático de informes diagnóstico por imágenes médicas, para el Reportador de la solución PACS del CESIM. El mismo fue desarrollado sobre la plataforma .NET con lenguaje de programación C#, utilizando el *Framework* 4.0. Se empleó la tecnología WPF y MVVM como patrón arquitectónico fundamental.

La validación del componente desarrollado se basó en el Modelo V, y arrojó que de 135 palabras médicas dictadas, 32 palabras mostraron dificultad pero con posibilidad de corregir o de añadir al diccionario, para un reconocimiento de voz de un 76,3 % de precisión.

### **Palabras claves:**

Informe diagnóstico, dictado automático.

# ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1. FUNDAMENTOS TEÓRICO METODOLÓGICOS RELACIONADOS CON EL RECONOCIMIENTO DE VOZ Y LOS INFORMES DIAGNÓSTICO</b> .....	<b>5</b>
1.1. RESEÑA HISTÓRICA DEL RECONOCIMIENTO DE VOZ. ....	5
1.2. CONCEPTOS FUNDAMENTALES ASOCIADOS A LA INVESTIGACIÓN. ....	6
1.3. BENEFICIOS DE LAS APLICACIONES QUE INCLUYEN EL RECONOCIMIENTO DE VOZ. ....	7
1.3.1. <i>Ventajas del reconocimiento de voz en las aplicaciones informáticas</i> . ....	7
1.3.2. <i>Desventajas del reconocimiento de voz en las aplicaciones informáticas</i> . ....	7
1.4. ARQUITECTURA BÁSICA DE UN SISTEMA DE RECONOCIMIENTO DE VOZ.....	8
1.5. SISTEMAS INFORMÁTICOS RELACIONADOS CON INFORMES DIAGNÓSTICO. ....	9
1.5.1. <i>Sistema de Almacenamiento, Transmisión y Visualización de Imágenes Médicas</i> . ....	9
1.5.2. <i>Sistemas para la edición de informes de estudios imagenológicos</i> . ....	10
1.6. VALORACIÓN DE SISTEMAS EXISTENTES QUE INCLUYEN EL RECONOCIMIENTO DE VOZ. ....	11
1.6.1. <i>Sistemas internacionales</i> .....	11
1.6.2. <i>Sistemas nacionales</i> . ....	14
1.7. TECNOLOGÍAS Y MODELOS DE CALIDAD EXISTENTES. ....	14
1.7.1. <i>Plataformas de desarrollo</i> . ....	14
1.7.2. <i>Bibliotecas para el reconocimiento de voz</i> . ....	16
1.7.3. <i>Herramientas de modelado</i> . ....	18
1.7.4. <i>Modelos de calidad de aplicaciones informáticas</i> . ....	19
<b>CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL COMPONENTE PARA REALIZAR EL DICTADO AUTOMÁTICO DE LOS INFORMES DIAGNÓSTICO</b> .....	<b>21</b>
2.1. MODELO DE DOMINIO. ....	21
2.2. DESCRIPCIÓN DEL PROCESO DE GENERACIÓN DE INFORMES DIAGNÓSTICO. ....	23
2.3. ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE. ....	23
2.3.1. <i>Requerimientos funcionales</i> . ....	23
2.3.2. <i>Requisitos no funcionales</i> . ....	24
2.4. DEFINICIÓN DE LOS ACTORES. ....	26



2.5.	DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	27
2.6.	DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA.....	27
2.7.	DISEÑO.....	28
2.8.	DESCRIPCIÓN DE LAS CLASES.....	29
2.9.	MODELO ARQUITECTÓNICO .....	29
<b>CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL COMPONENTE PARA REALIZAR EL DICTADO AUTOMÁTICO DE LOS INFORMES DIAGNÓSTICO.....</b>		<b>32</b>
3.1.	INTERFAZ DEL COMPONENTE.....	32
3.2.	ESTÁNDAR DE CODIFICACIÓN PARA EL COMPONENTE.....	33
3.2.1.	<i>Líneas y espacios en blanco.....</i>	33
3.2.2.	<i>Declaración de las clases.....</i>	33
3.2.3.	<i>Llaves.....</i>	33
3.2.4.	<i>Comentarios.....</i>	34
3.2.5.	<i>Tratamiento de errores.....</i>	34
3.3.	DIAGRAMA DE COMPONENTES .....	35
3.4.	DIAGRAMA DE DESPLIEGUE .....	36
3.5.	SEGURIDAD.....	38
3.6.	VALIDACIÓN DEL COMPONENTE DESARROLLADO.....	39
3.6.1.	<i>Pruebas unitarias.....</i>	40
3.6.2.	<i>Pruebas de integración.....</i>	41
3.6.3.	<i>Pruebas de sistema.....</i>	41
3.6.4.	<i>Pruebas de aceptación.....</i>	42
<b>CONCLUSIONES.....</b>		<b>44</b>
<b>RECOMENDACIONES.....</b>		<b>45</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>46</b>
<b>WEBGRAFÍA.....</b>		<b>51</b>
<b>BIBLIOGRAFÍA.....</b>		<b>54</b>
<b>ANEXOS .....</b>		<b>56</b>

ANEXO 1. PREFIJOS DE LOS REQUERIMIENTOS NO FUNCIONALES POR CATEGORÍA.....	56
ANEXO 2. REQUISITOS FUNCIONALES. ....	57
ANEXO 3 DIAGRAMA DE CLASES DE LOS CASOS DE USO ARQUITECTÓNICAMENTE SIGNIFICATIVOS. ....	58
ANEXO 4 DIAGRAMA DE SECUENCIA.....	58
ANEXO 5 DESCRIPCIÓN DE LAS CLASES. ....	59
ANEXO 6 TABLA CON LAS PALABRAS COMPROBADAS. ....	60

## **INTRODUCCIÓN**

El avanzado desarrollo de las tecnologías de la información y la comunicación (TIC), ha conducido a la sociedad a un nuevo milenio. Estas conforman un conjunto de recursos necesarios para manipular la información en los ordenadores, los programas informáticos y las redes necesarias para convertirla, almacenarla, administrarla, transmitirla y encontrarla.

La incorporación de las TIC al mundo sanitario ha evolucionado para mejorar la calidad de vida de los ciudadanos. Esto favorece el desarrollo de herramientas destinadas a dar respuestas en áreas como la planificación, la investigación, la gestión, prevención, promoción en el diagnóstico o tratamiento (1).

Cuba a pesar de ser un país bloqueado y de tener muchas limitaciones económicas ha dado grandes pasos en muchas ramas del sector social. Una de ellas es la medicina, la que posee actualmente equipos tecnológicos altamente desarrollados que permiten brindar mejor atención a la población. Además se encuentra inmersa en el campo de la informática con la elaboración de aplicaciones médicas que se integran a las altas tecnologías brindando así muchos servicios en los hospitales, entre los cuales se puede mencionar la radiología.

El rápido crecimiento en la radiología alcanzado por las técnicas formadoras de imágenes, ha proporcionado a una nueva ciencia más abarcadora, la imagenología. Esta se utiliza para revelar, diagnosticar y examinar enfermedades, o para estudiar la anatomía y las funciones del cuerpo (2). La radiología, la termografía médica, la endoscopia, la microscopía y la fotografía médica forman parte de esta técnica. Su principal ventaja es que permite obtener imágenes del cuerpo sin tener que abrirlo.

La generación de un gran cúmulo de imágenes, unido al desarrollo de la informática médica, las redes de comunicación, las computadoras, los medios de almacenamiento y servidores, permiten el surgimiento del Sistema de Almacenamiento y Transmisión de Imágenes Médicas (PACS, por sus siglas en inglés) y el Sistema de Información Radiológica (RIS, por sus siglas en inglés).

Los PACS (3) son los encargados de la adquisición, almacenamiento digital, visualización, transmisión y descargas de imágenes radiológicas. Tiene como ventajas la reducción del coste radiológico, de modo que elimina la necesidad de disponer y almacenar las placas radiológicas. Posibilita el acceso remoto a las imágenes, ayudando al radiólogo a optimizar su tiempo disponible. También proporciona una manera más rápida y confiable de acceder a los históricos de imágenes de un paciente.

# Componente para realizar el dictado automático en la elaboración de los informes diagnóstico. **Introducción**

---

El RIS (4) informatiza toda la actividad radiológica de un paciente, partiendo desde la petición del estudio al informe de este, pasando por la recogida de incidencias y consumo para la realización de dicha exploración. Un RIS posee la responsabilidad de gestionar la actividad clínica y administrativa del departamento imagenológico, manejar la información demográfica del paciente, programar las citas y la entrega de reportes diagnóstico.

La solución PACS del Centro de Informática Médica (CESIM), le ofrece al personal médico una gama de herramientas de propósito general, para la visualización y procesamiento de imágenes médicas y la posterior edición de los informes que son emitidos. Además facilita el acceso a las imágenes y los reportes diagnósticos desde cualquier punto de la institución hospitalaria. Está formado por varios subsistemas altamente integrados y compatibles con el estándar internacional Digital Imaging and Communication in Medicine en su versión 3.0 (DICOM).

Uno de los subsistema que lo conforman es el Reportador, que tiene como objetivo principal la confección de un reporte del estudio médico imagenológico realizado a un paciente para luego almacenarlo en el servidor RIS. Funciona en los diferentes servicios de imágenes y asistenciales de una institución de salud. Su modo de funcionamiento puede estar integrado al visor para los especialistas que generan sus propios reportes, o como una herramienta independiente, para los transcriutores que reescriben los reportes generados por los especialistas.

Actualmente existen dos flujos de trabajo básicos para la realización de informes diagnóstico, por transcripción y sin transcripción. Durante la transcripción existen retrasos en la entrega de un diagnóstico final al paciente. Esto se debe a que los transcriutores, que son los encargados de realizar las grabaciones médicas, no poseen un conocimiento médico en la mayoría de los casos y no entienden palabras dichas por los especialistas. Esto trae consigo que la información no se entregue con la calidad y el tiempo requerido. Además las grabaciones con mala calidad, también pueden generar atraso y un mal trabajo en los reportes finales.

Por otra parte la realización de informes sin transcripción pueden generar demora, al ser los especialistas los que editan el informe completamente, los que en ocasiones no poseen la agilidad y tiempo para su llenado digital. También trae consigo demora en la atención de un paciente a otro, debido que el médico requiere de tiempo para realizar el diagnóstico requerido para el mismo.

# Componente para realizar el dictado automático en la elaboración de los informes diagnóstico.

## Introducción

---

Por lo antes planteado, se identifica como **problema de la investigación**: ¿Cómo agilizar la gestión de la digitalización de los informes diagnóstico por imágenes médicas? Se define como **objeto de estudio**, los procesos de la digitalización de informes diagnóstico. Se enmarca en el **campo de acción** los procesos de la digitalización de informes diagnóstico mediante un dictado.

De acuerdo con la problemática planteada se propone como **objetivo general** desarrollar un componente que permita agilizar el dictado automático de informes diagnóstico por imágenes médicas para el Reportador de la solución PACS del CESIM.

Para dar cumplimiento al objetivo planteado se proponen las siguientes **tareas de la Investigación**:

- Realización de un análisis crítico y valorativo de los sistemas informáticos existentes de dictado automático a nivel nacional e internacional para los informes diagnóstico por imágenes.
- Realización de un análisis crítico y valorativo de las tendencias, técnicas, tecnologías, plataformas, metodologías y herramientas usadas en la actualidad.
- Elaboración de los artefactos correspondientes a las fases de desarrollo: “Modelamiento del Negocio”, “Requisitos”, “Análisis y Diseño” e “Implementación”, sirviendo de base a los desarrolladores.
- Implementación del componente aplicando las pautas de diseño y siguiendo lo establecido en la Especificación de Requisitos de Software.
- Integración del componente desarrollado al Reportador de la solución PACS del CESIM.

Los métodos de investigación utilizados se exponen a continuación:

### Métodos Teóricos

- Histórico-lógico para investigar y analizar los orígenes y evolución de las herramientas para el reconocimiento de voz, con el objetivo de lograr una mejor comprensión del objeto y el campo de estudio propuesto.
- Análisis-sintáctico en el estudio de las principales técnicas y herramientas existentes para la creación de un componente para realizar el dictado automático de los informes diagnóstico.
- Inducción-deducción para el análisis y la definición de la estrategia a utilizar para el desarrollo del componente a partir de los problemas detectados.

### Método Empírico

# Componente para realizar el dictado automático en la elaboración de los informes diagnóstico.

## Introducción

---

- Observación directa de la realización de los informes diagnóstico, permitió tener un análisis del problema en cuestión.

El presente documento sigue una estructura como se muestra a continuación:

### **Estructuración del contenido**

Capítulo 1: Contiene un estudio del estado del arte de las principales herramientas que existen en el mundo para el dictado automático para los informes diagnóstico. Se realiza un análisis de las diferentes metodologías y tecnologías que pueden ser utilizadas en el desarrollo de la solución y la selección de las más apropiadas para llevar a cabo dicho proceso con calidad en el desarrollo de la investigación.

Capítulo 2: Contiene lo referente al análisis y diseño del sistema, se describen las características del sistema, se presenta el modelo del negocio; así como el glosario, elaborado con los principales conceptos del negocio y las relaciones que se establecen entre ellos. Se realiza la especificación de los requisitos funcionales y no funcionales del sistema, determinándose a su vez los casos de uso y los actores. Se describe cada uno de los casos de uso. Como parte de la solución se modelan los diagramas de clases del análisis, los diagramas de clases del diseño, así como los diagramas de interacción correspondientes. Se exponen los patrones de diseño empleados en la solución y se muestra la estructura del sistema a través de la arquitectura del mismo.

Capítulo 3: Aborda todo lo relacionado con el flujo de trabajo de implementación y prueba del sistema. En el mismo se especifican los componentes ejecutables del sistema, así como su interacción. Se exponen además los diagramas de implementación y despliegue; artefactos generados en esta fase de construcción del sistema.

## CAPÍTULO 1. FUNDAMENTOS TEÓRICO METODOLÓGICOS RELACIONADOS CON EL RECONOCIMIENTO DE VOZ Y LOS INFORMES DIAGNÓSTICO.

En este capítulo se realiza un análisis del estado del arte de sistemas similares utilizados a nivel internacional y a nivel nacional, que suponen una solución al problema de la presente investigación. También se analizan las principales herramientas y tecnologías actuales, usadas para el desarrollo del sistema, así como la metodología de desarrollo y el lenguaje de modelado a utilizar. Todo esto con el objetivo fundamental de brindar información complementaria para propiciar un mejor entendimiento de los capítulos venideros.

### 1.1. Reseña histórica del reconocimiento de voz.

La historia de la investigación en el campo del reconocimiento de voz se ha evidenciado con más fuerza a partir de la segunda mitad del siglo XX (5) hasta nuestros días. Durante este período se han construido máquinas y sistemas que abarcan, desde el reconocimiento de pequeños grupos de palabras, hasta el dictado para grandes vocabularios.

El avance de la informática fue uno de los elementos que posibilitó la mejora y evolución de los sistemas de reconocimiento de voz. Anterior a la misma, solo existían dispositivos mecánicos y electrónicos, tal es el caso de la máquina propuesta por J.B. Flowers (6) en 1916, que podía convertir los sonidos de las letras en ondas de la misma naturaleza que las que se transmitían entre los submarinos, a las que llamó alfabeto fonográfico. La máquina consistía en dos electroimanes y condensadores ajustados para cubrir todo el rango de frecuencias.

En los años cuarenta se desarrolló un dispositivo capaz de visualizar la señal acústica sobre papel, conocido como espectrógrafo. Este permitía la obtención de un registro de la energía contenida en las diversas bandas de frecuencia de una palabra o frase en función del tiempo, llamado espectrograma. Otro de los dispositivos electrónicos que tuvo gran impacto en la época, fue el creado en 1952 por Laboratorios Bell (6), que era capaz de discriminar con cierta precisión diez dígitos pronunciados de forma aislada por un único locutor.

Otro trabajo paralelo e independiente se desarrolló en los Laboratorios RCA en 1956 (7). H.F. Olson y H. Bellar desarrollaron una máquina de escribir fonética, donde la señal obtenida del micrófono pasaba a través de un amplificador y un compresor antes de ser aplicada a un banco de 8 filtros. Esta información se pasaba

a un banco de reguladores mediante un interruptor que actuaba cada 40 ms desde el momento en que la pronunciación comenzaba. Al final se obtenía un dibujo característico para los distintos sonidos.

Uno de los primeros experimentos de reconocimiento de voz realizados en ordenadores, fue a principio de los años 60 cuando P. Deves y M. V. Mathews construyeron un reconocedor de dígitos basado en un computador IBM 704 (6). Al mismo tiempo T. K. Vintsyuk (8), propone la utilización de métodos de programación dinámica para conseguir el alineamiento temporal de pares de realizaciones de habla.

Durante la década de los 70 se dieron grandes pasos de avance, con la incorporación de técnicas como "Time warping", "Modelado probabilístico" (Aplicación de los Modelos ocultos de Markov), y el "Algoritmo de Retropropagación" (9). Estas técnicas posibilitaron la disminución de los costos de las aplicaciones, y el incremento de los vocabularios.

A principio de los 80 se realizó un gran trabajo en desarrollar sistemas capaces de extraer conocimiento de forma inductiva, es decir, a partir de muestras. Se siguen desde entonces los trabajos de IBM, una modelización acústica basada en Modelos Ocultos de Markov, discretos y continuos, y se optiman los algoritmos de aprendizaje para entrenar los sistemas a partir de grandes bases de datos. Se desarrollaron algoritmos de búsqueda eficientes con los que determinar la secuencia óptima de patrones para una secuencia de vectores acústicos.

A lo largo de toda esta etapa evolutiva del reconocimiento de voz, fueron varias las empresas de distintos países que incursionaron en su investigación. Las necesidades de uso de este campo se presentan en entornos industriales, en la medicina, trabajo en casa, disminución de costes de transporte y almacenamiento de datos, tele-reuniones, acceso rápido e integrado a bloques selectivos de información, telefonía móvil y otros.

En la actualidad los esfuerzos investigativos se concentran en mejorar el rendimiento de la generación automática de unidades acústicas contextuales, el entrenamiento discriminativo de los modelos y de estrategias de búsqueda heurísticas en grandes autómatas que representan modelos del lenguaje (6). También se está realizando una gran labor en diseñar y adquirir grandes bases de datos para el entrenamiento de sistemas de reconocimiento de discurso continuo.

## 1.2. Conceptos fundamentales asociados a la investigación.

A continuación se conceptualizan los términos fundamentales asociados a la investigación:



**Reconocimiento de voz:** Es la capacidad de un ordenador, de convertir, las palabras de la voz humana a un código binario comprensible por la computadora. Esto lo realiza mediante el método de comparar las palabras habladas con grabaciones previas de una base de datos en función de diferentes parámetros.

**Informe diagnóstico:** Es la evaluación detallada que realiza el médico sobre el estudio imagenológico realizado a un paciente. Posee informaciones como los datos demográficos y clínicos del paciente, el tipo de estudio realizado, descripción de los hallazgos y el diagnóstico realizado sobre estos.

### 1.3. Beneficios de las aplicaciones que incluyen el reconocimiento de voz.

La incorporación del reconocimiento de voz en las aplicaciones informáticas, se ha convertido en una herramienta muy usada a nivel mundial para introducir textos en un ordenador. Este ofrece una serie de ventajas y desventajas para los usuarios que las utilizan en las distintas esferas sociales.

#### 1.3.1. *Ventajas del reconocimiento de voz en las aplicaciones informáticas.*

La utilización del reconocimiento de voz en las aplicaciones informáticas proporciona varias ventajas. Un ejemplo claro es su uso por las personas que presentan alguna discapacidad física en las extremidades superiores, al posibilitar que los mismos puedan interactuar con el ordenador a través de comandos.

En el área de la telefonía móvil tiene gran significación, pues hace posible una interacción entre las personas y estos dispositivos, llegando a ser estos últimos asistentes virtuales. Permite la búsqueda de información dentro de los propios celulares o en *Internet*. Simplifica las acciones básicas como realizar llamadas, escribir mensajes y correos, y tomar notas.

Uno de los sectores que más se beneficia con el reconocimiento de voz es la salud. Permite a los médicos disminuir el tiempo de la realización de informes diagnóstico, historias clínicas e informes de otro tipo. También posibilita optimizar la gestión del flujo de trabajo, y aumentar el número de pacientes consultado por cada especialista, reduciendo su tiempo de estancia en el hospital.

#### 1.3.2. *Desventajas del reconocimiento de voz en las aplicaciones informáticas.*

Una de las desventajas que trae consigo el uso del reconocimiento de voz en las aplicaciones informáticas es la imprecisión y la lentitud debido a la revisión y corrección requeridas después de dictarle un documento al ordenador. Aunque la mayoría de las veces el programa de reconocimiento de voz puede interpretar las palabras habladas correctamente, igualmente se necesita realizar correcciones en cuanto a la puntuación.

Además, el programa podría no reconocer palabras tales como nombres de marcas o sobre nombres no comunes hasta que hayan sido agregados a la biblioteca de palabras del programa.

También tiene como desventaja los factores ambientales, pues el ambiente ideal para un programa de reconocimiento de voz es uno tranquilo, especialmente si no se utiliza un micrófono que filtre el sonido. En un contexto ruidoso, este método podría fallar, incluso podría generar texto de voces que se escuchan detrás. No obstante, en la actualidad existen aplicaciones de este tipo que son tolerantes al ruido.

De igual manera la poca fluidez en el habla de las personas o el poco entrenamiento del sistema, puede afectar el entendimiento de las palabras, y específicamente en el caso de los reconocedores de voz de propósito general, el vocabulario puede ser extenso.

## 1.4. Arquitectura básica de un sistema de reconocimiento de voz.

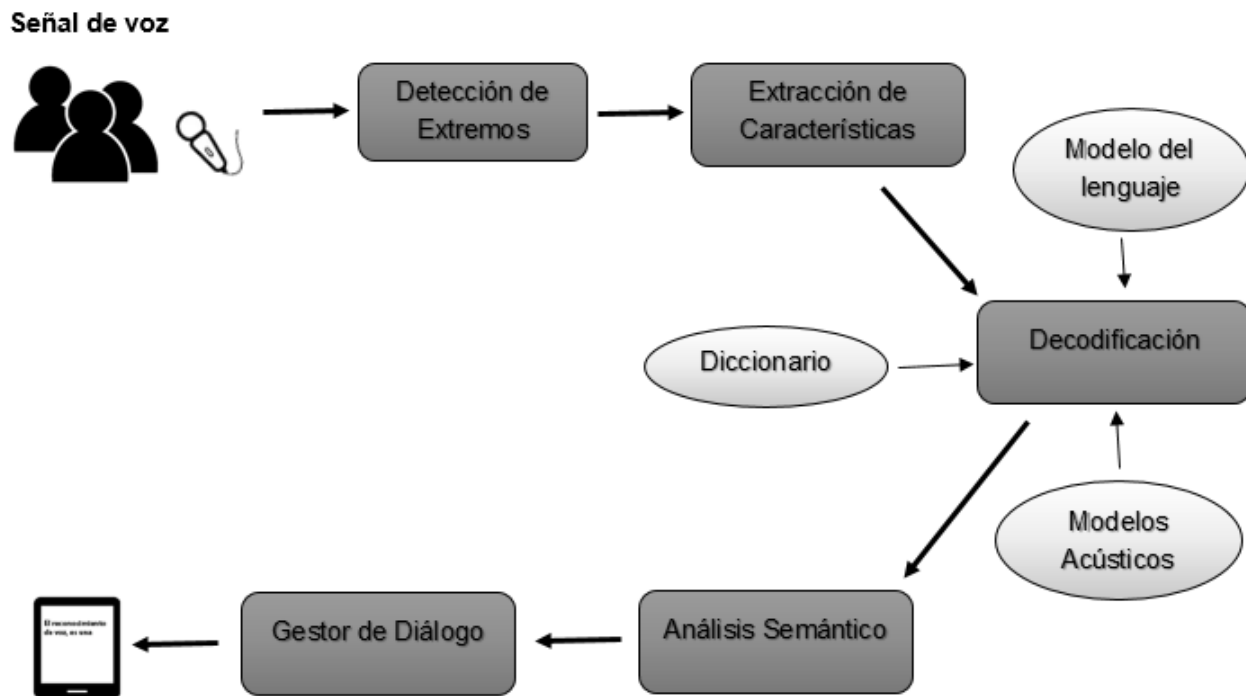


Fig. 1 Modelo genérico de un sistema de diálogo (9).

En la figura 1 se aprecia que la señal de voz es capturada por un micrófono y digitalizada, para luego pasar al “Detector de extremos” el cual detectará la presencia de voz y la entregará al próximo bloque. La “Extracción de características” consiste en dividir la señal en varios segmentos, obteniendo así una representación de las características acústicas más distintivas de cada segmento. A partir de estas características se construyen vectores que son la entrada del siguiente módulo.

En el “Decodificador” se genera la frase reconocida haciendo uso de modelos acústicos, que proporcionan información sobre características y propiedades de los sonidos identificados; modelos de lenguaje, que contienen información sobre cómo deben combinarse las palabras para formar frases; y del diccionario, que indican el conjunto de sonidos que forman las palabras del vocabulario. Una segunda etapa comienza con el módulo “Análisis semántico”. Este módulo se encarga de extraer un significado coherente de la frase, su salida la entrega al “Gestor de diálogo” que hará una predicción de la próxima interacción con el usuario en función del estado actual del diálogo.

## 1.5. Sistemas informáticos relacionados con informes diagnóstico.

A continuación se explica el funcionamiento de sistemas informáticos que están estrechamente relacionados con la presente investigación, específicamente contienen la funcionalidad de generación de informes diagnóstico.

### 1.5.1. Sistema de Almacenamiento, Transmisión y Visualización de Imágenes Médicas.

Los sistemas PACS (10) ofrecen una alternativa en el manejo de imágenes digitales en forma eficiente a través de dispositivos conectados en una red, permiten proveer servicios de almacenamiento, tratamiento y transferencia de información, para dar soporte a las áreas donde se genera un volumen importante de imágenes.

Su tarea fundamental es mantener el flujo de las imágenes adquiridas en las diferentes modalidades. También es el responsable de la adquisición de imágenes con alta resolución, asegura tener una gran capacidad de almacenamiento para dar soporte a la actividad imagenológica. Otras características serían la posibilidad de centralizar el almacenamiento y diagnosticar en cualquier computadora que se encuentre conectada a la red de imágenes. En esta se pueden observar estudios de distintas modalidades, siempre y cuando posea un visor de imágenes médicas (11).

Con el objetivo de que estos sistemas funcionen correctamente con modalidades y estaciones de trabajo de diferentes fabricantes, existen una serie de estándares de imagen digital que se han definido para ello. Todos estos sistemas, modalidades y estaciones de trabajo que se comuniquen entre sí, deben hacerlo bajo el estándar DICOM. Este es el estándar para imagen médica digital, y el formato universal para el intercambio de imagen médica digital.

El desarrollo de los PACS (12) trajo consigo transformaciones en el esquema de funcionamiento de los departamentos de diagnósticos por imágenes, tales como el reemplazo de las llamadas placas radiológicas por el almacenamiento de las imágenes en discos duros, introdujo la tele consulta y permitió la inserción de la educación a distancia en el área imagenológica, mejorando significativamente la eficiencia y calidad de la atención que se les brinda a los pacientes.

### *1.5.2. Sistemas para la edición de informes de estudios imagenológicos.*

El Reportador (13) constituye uno de los módulos del sistema PACS desarrollado por el Departamento de Software Médico Imagenológico del CESIM. Este módulo posee un repositorio centralizado al que se puede acceder desde cualquier otra estación de reporte, desde el Reportador o desde el sistema RIS. Además el sistema permite la codificación de los hallazgos con el uso de la Clasificación Internacional de Enfermedades CIE10.

Existen dos flujos de trabajo básicos para la realización de informes diagnósticos en el Reportador:

- ✓ Por transcripción: el especialista realiza un dictado de los hallazgos y los comentarios del estudio y el paciente que considere. Una vez lista la grabación, emite el informe llegándole este al transcriptor. Una vez que se transcribe dicho informe, se lo envía de regreso al especialista, para que sea revisado y aprobado. Si el especialista no está de acuerdo con la transcripción, puede rechazar el informe, regresándolo al transcriptor, aunque también podría corregir los defectos él mismo. Una vez que el informe queda listo, el especialista lo aprueba y éste se almacena en los servidores del hospital, y entonces podrá ser visualizado en cualquiera de las estaciones de la red de imágenes.
- ✓ Sin transcripción: el especialista edita el informe completamente y una vez que está listo lo aprueba; almacenándose éste en los servidores del hospital. Una vez allí, dicho informe puede ser visualizado en cualquiera de las estaciones de trabajo de la red de imágenes.

Brinda la posibilidad de realizar la transcripción mediante el grabado de voz y agiliza el proceso de confección del documento con la obtención automática de la información contenida en el fichero DICOM. A pesar de esto, presenta una serie de privaciones e insuficiencias que limitan su funcionamiento y entorpecen el proceso de emisión de informes diagnósticos.

## 1.6. Valoración de sistemas existentes que incluyen el reconocimiento de voz.

Un sistema de reconocimiento de voz está diseñado para agilizar las labores de interpretación de radiología y minimizar el tiempo y esfuerzo requerido para completar las actividades de reporte diarias. A continuación se describen algunos de estos sistemas identificados según la literatura consultada.

### 1.6.1. *Sistemas internacionales.*

**Dragon NaturallySpeaking 8 Preferred Edition Spanish36 (9):** es una herramienta, desarrollada por Scansoft® para reconocimiento y síntesis de voz. Se integra a las aplicaciones de los sistemas operativos Microsoft Windows XP y Microsoft Windows 2000. No es un *kit* de desarrollo, pero sus motores son compatibles con Microsoft SAPI.

Sus características principales son:

- ✓ Reconocimiento con 99% de precisión.
- ✓ Reconocimiento de 160 palabras/minuto.
- ✓ Creación de comandos personalizados.
- ✓ Vocabulario expansible de 300.000 palabras.
- ✓ Edición y formato de cualquier texto por comandos de voz.
- ✓ Compatible con sistemas Pocket PC.
- ✓ Inicio y control de cualquier aplicación Microsoft Windows como: Microsoft Word, Microsoft Excel y Microsoft Internet Explorer.
- ✓ Soporte para reconocimiento en inglés y español.

El anterior sistema es bueno por las razones antes mencionadas, pero presenta algunas deficiencias cuando se dicta una palabra que no se encuentra en el vocabulario del sistema, el programa no podrá transcribirla. Además presenta problemas de compatibilidad con Windows Vista, solo puede tener instalada una versión del sistema.

La licencia se proporciona a título individual. Puede instalar el programa en más de un equipo (por ejemplo, en uno de escritorio y en uno portátil, o en uno del trabajo y otro de su domicilio), pero no podrá usarlo simultáneamente en más de un equipo.

Se puede crear varios perfiles de voz, siempre que todos ellos sean de la misma persona. Si otra persona desea crear un perfil de voz, esa persona tendrá que adquirir otra licencia del sistema (14).

**Fonix Voiceln 4.0.1 (9):** fue desarrollado en base a redes neuronales, y a diferencia de otros sistemas no requiere entrenamiento por parte del usuario.

Entre sus características destacan:

- ✓ Soporte para caracteres Unicode.
- ✓ Tolerante a ruido y a ambientes diversos.
- ✓ Independiente del locutor.
- ✓ Gramáticas dinámicas.
- ✓ Soporte para Visual Basic, C# y Java.
- ✓ Soporte para Microsoft Windows en sus versiones CE, 98, 2000 y XP; además de Linux, Solaris, MAC OS, Palm OS y Symbian.

**Sistema QdReport (15):** es un módulo de transcripción de informes médicos de Agfa HealthCare. QdReport optimiza la productividad de los médicos y también de los transcritores, lo que genera tiempos de respuesta más rápidos y menores costos. Además incorpora los sistemas de procesamiento de voz perfectamente para optimizar el dictado digital y el reconocimiento de voz, y puede emplearse como una solución independiente o integrada con otras soluciones de aplicaciones.

Características del sistema:

- ✓ Excelentes características de procesamiento de texto.
- ✓ Dictado digital y reconocimiento de voz.
- ✓ Generación de informes de administración.
- ✓ Protección segura de los datos.

**Sistema SpeechMagic (15):** es un módulo incorporado en el sistema de información radiológica QDoc y el módulo de transcripción de informes médicos QdReport de Agfa HealthCare. Con este sistema accede a tecnología innovadora de voz incorporada en las aplicaciones para generación de informes médicos de Agfa

HealthCare. Al trabajar en conjunto con Nuance Healthcare Solutions, logran que el dictado digital y el reconocimiento de voz sean flexibles, sumamente exactos y muy fáciles de manejar para todos los usuarios que la utilizan.

Otras características del sistema:

- ✓ El dictado digital asegura un flujo de trabajo óptimo.
- ✓ Flexible y adaptable.
- ✓ El reconocimiento de voz transforma la palabra en texto.
- ✓ Mayor eficiencia para el informe de resultados.
- ✓ Un trabajo más satisfactorio para el personal administrativo.
- ✓ Servicio mejorado para los médicos solicitantes.

**SpeechExec de Philips (16):** la aplicación puede ofrecer hasta un 99 % de tasa de reconocimiento, pero sin la automatización del flujo de documentos entre autores y asistentes, todas las ventajas desaparecen. Utiliza mucho tiempo en importar los dictados al sistema. Para su uso solo hay que conectar su grabadora de dictado en la base y la aplicación de reconocimiento de voz descarga y transcribe automáticamente sus dictados.

SpeechExec se puede integrar completamente en el flujo de trabajo de una oficina y ofrece funciones esenciales requeridas, como colas automáticas de dictados de varios autores para asistentes predefinidos, utilización del reconocimiento de voz en formularios y plantillas, enrutamiento selectivo de los archivos de dictado al sistema de reconocimiento de voz o a los asistentes.

**IBM ViaVoice (17):** este sistema incluye un vocabulario de 25.000 palabras de la especialidad. Permite personalizar las plantillas de texto y puede integrarse en cualquier aplicación gracias a la utilización de controles Active-X y una interfaz con HL-7. Este es el único producto que incluye un vocabulario médico en español, de 38.000 vocablos, por un precio aproximado de 34.000 pesetas. Además permite un control total del ordenador (excepto los cuadros de diálogo). Su exactitud es similar al resto, de un 95%, y el tiempo de entrenamiento estimado es de 30 minutos.

La principal desventaja de estos sistemas radica en los altos costos de sus productos y del soporte técnico, incluyendo la capacitación para el uso del sistema, lo cual constituye una barrera principalmente para los países del tercer mundo. Además por ser de licencias privativas no se pueden reutilizar sus funcionalidades

y no brindan mucha información sobre cómo se realizan. También el estudio de estos sistemas permitió aumentar el conocimiento del funcionamiento del reconocimiento de voz sirviendo de guía para el desarrollo de la solución propuesta.

## 1.6.2. *Sistemas nacionales.*

Luego de haber realizado un estudio en Cuba se determinó que no existen aplicaciones que realicen esta función por eso el Departamento de Software Médico Imagenológico (SWMI) de la Universidad de las Ciencias Informáticas (UCI) ha asumido la tarea de desarrollar un componente que permita redactar los informes de diagnósticos por imágenes de forma automática mediante el dictado que realizan los médicos, ahorrando así, en tiempo, calidad, y evitando errores en un diagnóstico final.

A partir del estudio realizado sobre los sistemas de reconocimiento de voz se determinó que no se pueden utilizar en la solución del componente, porque se encontró que la mayoría de estos sistemas son comercializados por compañías de aplicaciones privadas y estos a su vez mayoritariamente se integran al Reportador, estas compañías como parte de su estrategia brindan muy poca información sobre los mismos. Presentan varios problemas con un reconocimiento pobre de la voz, dificultad para detener la transcripción, comprobación de la ortografía, y otros problemas de funcionamiento con la interface del sistema con el RIS. Su principal desventaja radica en los altos costos de sus productos y del soporte técnico incluyendo la capacitación para el uso del sistema, lo cual constituye una barrera principalmente para los países del tercer mundo.

## 1.7. Tecnologías y modelos de calidad existentes.

Para la realización del componente para el reconocimiento de voz se utilizaron las tecnologías y lenguajes de programación definidas por el departamento de SWMI del CESIM y además se incluyeron otras necesarias para su desarrollo. A continuación se hace referencia a sus principales características.

### 1.7.1. *Plataformas de desarrollo.*

Actualmente existen varias plataformas de desarrollo de aplicaciones informáticas, tal es el caso de la plataforma Java EE (18) que por su robustez dota a los sistemas que la incluyen de un carácter empresarial. También existe la plataforma de Oracle (19) que provee una gama completa e integrada de herramientas para el desarrollo de aplicaciones, desarrollo de base de datos e inteligencia de negocio. Otras de las



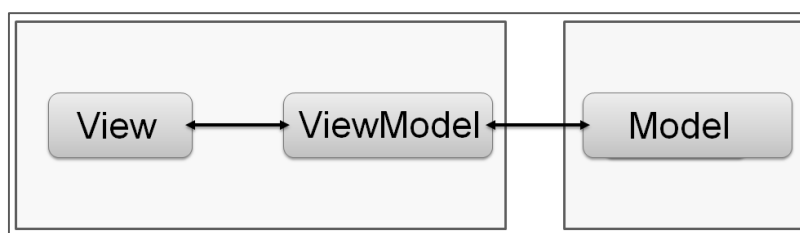
plataformas es la de .NET (20) que es un componente integral de Windows que admite la compilación y la ejecución de la siguiente generación de aplicaciones y servicios Web XML.

Además .NET proporciona un entorno coherente de Programación Orientada a Objetos (POO por sus siglas en inglés) de ejecución de código, que reduce lo máximo posible la implementación de aplicaciones y los conflictos de versiones. También basa toda la comunicación en estándares del sector para asegurar que el código de la plataforma se pueda integrar con otros tipos de código.

De las plataformas antes mencionadas se identifica la de .NET como la ideal para ser utilizada en este trabajo debido a que las potencialidades que ofrece se ajustan a la presente investigación. Además en el CESIM se tiene gran experiencia con dicha plataforma y es su propuesta para el desarrollo de los sistemas enmarcados en el departamento de SWMI correspondiente a dicho centro.

Uno de los componentes de esta plataforma es Windows Presentation Foundation (21) (WPF por sus siglas en inglés), la cual proporciona a los desarrolladores un modelo de programación unificado con el que se generan experiencias de clientes inteligentes de Windows, en las que se incorporan interfaz de usuario, multimedia y documentos. Como patrón arquitectónico para la creación de aplicaciones WPF se va a utilizar Model View ViewModel (22) (MVVM por sus siglas en inglés) que es una herramienta eficaz para crear soluciones y un lenguaje común para analizar el diseño de la aplicación con los desarrolladores.

Este patrón es una variante del Modelo Vista Controlador que separa el modelo y la vista (interfaces de usuario) introduciendo una capa entre ellos que sirve de enlace llamada modelo de vista (ViewModel). El modelo contiene el dominio del negocio: reglas del negocio, acceso a datos, clases modelo; la vista está formada por las interfaces de usuario y el ViewModel es responsable de agregar y almacenar información que será enviada a la vista. Los dos objetivos principales de ViewModel son: hacer que Model sea fácil de adoptar por WPF/XAML View, y separar y encapsular el Model desde la View.



*Fig.2 Esquema general del patrón Model View ViewModel.*

La plataforma .NET utiliza varios lenguajes de programación, dentro de ellos se decidió utilizar C# 4.0 (23) que mejora su interoperabilidad con otros lenguajes y tecnologías. Entre sus características se pueden mencionar la portabilidad del código, no admite funciones ni variables globales sino que todo el código y los datos han de especificarse dentro de definiciones de tipos de datos, consta de un editor de código completo, plantillas de proyecto, asistentes para código, así como un depurador muy eficiente y fácil de usar.

Como entorno de desarrollo integrado (IDE por sus siglas en inglés) se seleccionó Microsoft Visual Studio 2010 (24) que es un conjunto de herramientas de desarrollo para construir aplicaciones Web ASP.NET, XML Web Services y aplicaciones de escritorio. Utiliza la tecnología WPF y lenguajes como Visual Basic, Visual C++ y Visual C# son soportados por este IDE. Permite realizar una administración del ciclo de vida de las aplicaciones e incorpora otras de pruebas que ayudan a garantizar la calidad del código en todo momento. Es compatible con plataformas como Windows, Windows Server, Web, Cloud, Office y SharePoint, entre otras, todo en un único IDE.

### 1.7.2. Bibliotecas para el reconocimiento de voz.

En la actualidad existen varias librerías e Interfaces de Programación de Aplicaciones (API por sus siglas en inglés) para realizar el reconocimiento de voz, tal es el caso de Java Speech (JSAPI) (25) que define una interfaz de aplicación multiplataforma estándar y fácil de usar para implementar la tecnología de habla en java. También existe Microsoft Speech Application Program Interface (26) (SAPI) 5.1 que constituye una capa entre el *hardware* (captura y reproducción) y los motores de reconocimiento y síntesis instalados en el sistema operativo.

Las bibliotecas de SAPI son gratuitas y descargables directamente desde el sitio de Microsoft, están implementadas en C++. Sus objetos permiten crear aplicaciones de voz dependientes del locutor con una interpretación correcta de hasta un 95% según el entrenamiento.

Características de SAPI:

- Tiempo de proceso de reconocimiento ajustable, esto se denomina como *Complete-Phrase Value*.
- Niveles de aceptación y rechazo ajustables.
- Síntesis de voz para los idiomas inglés norteamericano, y español.
- Generación de archivos WAV de 16 bits a 22Khz monoaural.
- Velocidad, volumen y tono de reproducción configurables.

- Soporte XML, con lo que se logra reutilizar componentes, crear listas de frases, y ajustar un correcto sentido semántico a los distintos niveles de la gramática.
- Soporte Unicode.
- Compatible con Microsoft Visual C++ y Microsoft Visual Basic.
- Dependiente del locutor.
- Reconocimiento de voz para los idiomas inglés norteamericano, chino y japonés.
- Compatible con Microsoft Windows.

La arquitectura SAPI está compuesta de 8 componentes básicos:

- Comandos de voz (alto-nivel): comandos precisos e identificables por la aplicación como una orden o respuesta a un evento.
- Dictado continuo (alto-nivel): componente que se encarga de escribir, lo hablado por el locutor, en cuadros de texto. Da formato a lo reconocido.
- Texto a voz simple (alto-nivel): funciones básicas para reproducir texto, de una aplicación basada en Windows, hacia un dispositivo de audio.
- Controles telefónicos (alto-nivel): componentes gráficos que contienen subrutinas aplicables a sistemas de síntesis y reconocimiento telefónico.
- Objetos compartidos: permite usar los motores de síntesis y reconocimiento a los objetos de alto nivel.
- Reconocimiento de voz directo (bajo-nivel): componentes que tienen acceso directo a los motores de reconocimiento, proveen mayor control.
- Síntesis de voz directa (bajo-nivel): componentes con acceso directo a los motores de síntesis de voz.
- Objetos destinos y objetos fuentes de audio (bajo-nivel): objetos que permiten a los desarrolladores acceder a los motores y utilizarlos en sus aplicaciones.

SAPI incluye tres tipos de gramáticas:

- Gramáticas libres de contexto: tienen reglas que predicen las próximas palabras que se hablarán.
- Gramáticas de dictado: definen un contexto para el locutor considerando el tema en cuestión, idioma y dictados satisfactorios anteriores.

- Gramáticas de dominio limitado: no proveen una estructura sintáctica rígida, pero si provee una amplia gama de palabras disponibles.

SAPI utiliza una librería de interoperabilidad SpeechLib.dll la cual es necesaria para interactuar con el reconocimiento de voz del sistema operativo Windows, la misma se usa para el funcionamiento del componente.

### 1.7.3. Herramientas de modelado.

En la actualidad existen varias herramientas de modelado, tal es el caso Visual Paradigm (27) para UML que genera documentación del sistema que se está desarrollando en formato PDF, HTML y Word, también permite la generación de código a partir de diagramas. También existe la herramienta Enterprise Architect (28) (EA) en su versión 7.5, que es para el diseño y análisis UML, que cubre todos los aspectos del ciclo de desarrollo de la aplicación; a través de las fases de requerimiento, análisis, diseño, prueba, control de cambios e implementación con una trazabilidad completa.

EA ayuda a manejar grandes modelos y complejas informaciones; permite compartir modelos o aspectos específicos de estos; proporciona una poderosa generación y reporte de documentos con editor de plantillas WYSIWYG para Formato de Texto Enriquecido (RTF por sus siglas en inglés) o HTML, soporta la ingeniería inversa de código fuente para lenguajes como C++, C#, Java, Delphi, VB.Net, Visual Basic, ActionScript, Python y PHP; así como para sistemas DBMS como Oracle 9i, 10g o 11g; SQL Server; MySQL; Access 2007 y PostgreSQL.

De las herramientas antes mencionadas se selecciona EA como la ideal para ser utilizada en este trabajo debido a que las potencialidades que ofrece se ajustan a la presente investigación. Además en el CESIM se tiene gran experiencia con dicha plataforma y es su propuesta para el desarrollo de los sistemas enmarcados en el departamento de SWMI correspondiente a dicho centro.

Como metodología de desarrollo se va a utilizar Proceso Unificado de Desarrollo (29) (RUP por sus siglas en inglés), utiliza UML para representar los esquemas de una aplicación. Divide en 4 fases el desarrollo de la aplicación: inicio, elaboración, construcción y transición. Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Entre las características principales de RUP se encuentran: dirigido por casos de uso, iterativo e incremental y centrado en la arquitectura.

También se seleccionó el Lenguaje Unificado de Modelado (30) (UML por sus siglas en inglés), es por naturaleza el más utilizado para lenguajes orientados a objetos como C++, Java y C#. UML es independiente de la metodología que se desee utilizar, lo que ha provocado sin dudas el amplio uso que tiene en la industria de desarrollo de aplicaciones.

UML (31) se considera lo suficientemente expresivo para modelar otros elementos como flujos de trabajo, diseño de *hardware*, entre otros. Es un lenguaje gráfico para visualizar, especificar, describir métodos o procesos, construir y documentar un sistema informático. Se utiliza para dar soporte a metodologías de desarrollo de aplicaciones ágiles o robustas.

#### *1.7.4. Modelos de calidad de aplicaciones informáticas.*

Actualmente existen varios modelos de calidad de aplicaciones, tal es el caso del Modelo ISO/ IEC 15504 (32) que su principal objetivo es la generación de métricas de proceso y de producto para dar soporte a la planificación efectiva y así mejorar la calidad de los productos. También existe el Modelo IT Mark (32) que es un servicio internacional de certificación que estudia los procesos técnicos y de negocio, para medir el reconocimiento de Excelencia en Tecnologías de la Información. Otro es el Modelo de Capacidad y Madurez (29) (CMMI por sus siglas en inglés), es un conjunto de modelos elaborados por el Instituto de Ingeniería del Software (SEI por sus siglas en inglés) para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de cómputo.

CMMI es un marco de referencia que las organizaciones pueden emplear para mejorar sus procesos de desarrollo, adquisición, y mantenimiento de productos y servicios. Este modelo proporciona una guía para desarrollar procesos, que además ayuda a evaluar la madurez de la organización o capacidad de un área de procesos. Contiene 22 áreas de procesos (PA por sus siglas en inglés). Cada PA está formada por objetivos específicos y genéricos, prácticas específicas y genéricas. Puede representarse de forma continua y escalonada. La representación continua está formada por 5 niveles de capacidad del proceso y la escalonada por 5 niveles de madurez de procesos.

De los modelos de calidad mencionados se identifica CMMI como el ideal para utilizar, debido que CESIM certificó el nivel 2 de los 5 niveles de la representación escalonada. Este nivel consta de 7 PA:

Administración de Configuración (CM por sus siglas en inglés), Medición y Análisis (MA por sus siglas en inglés), Monitoreo y Control del Proyecto (PMC por sus siglas en inglés), Planeación de Proyecto (PP por sus siglas en inglés), Aseguramiento de la Calidad a Proceso y Producto (PPQA por sus siglas en inglés), Administración de Requisitos (REQM por sus siglas en inglés) y Administración de Acuerdo con Proveedores (SAM por sus siglas en inglés).

## **Consideraciones del Capítulo.**

Mediante el presente capítulo se pudo comprender el funcionamiento y la arquitectura básica del reconocimiento de voz. Además se expone el estudio realizado sobre algunos de los sistemas que poseen la funcionalidad de dictado automático, lo cual arrojó que no son una solución factible para dar solución a la problemática de la investigación planteada.

El estudio sobre las bibliotecas existentes para el reconocimiento de voz, permitió la selección de la tecnología Microsoft Speech Application Program Interface (SAPI) 5.1 con la librería de interoperabilidad SpeechLib.dll por ser esta gratuita e integrarse con facilidad al entorno del sistema Reportador del CESIM. Las herramientas a utilizar para el desarrollo del componente, favorecen la rápida implementación por tener una alta curva de aprendizaje, además son totalmente compatibles con el ambiente de trabajo del departamento SWMI del CESIM.

## **CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL COMPONENTE PARA REALIZAR EL DICTADO AUTOMÁTICO DE LOS INFORMES DIAGNÓSTICO.**

En el presente capítulo se representa los conceptos significativos en el dominio del problema. Además se especifican, describen y modelan los requisitos funcionales y no funcionales; así como los casos de uso del sistema identificados. También los diagramas de clase de diseño y de secuencia que participan en la relación con los casos de uso. Se brinda una breve descripción de las clases involucradas en la realización de los casos de uso arquitectónicamente significativos, y se describe el estilo arquitectónico que se utiliza en la estructura del sistema.

### **2.1. Modelo de dominio.**

Un Modelo de Dominio (33) es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea construcción del modelo de dominio. Se presenta como uno o más diagramas de clases, que contiene no conceptos propios de un sistema de aplicaciones sino de la propia realidad física.

Los modelos de dominio (34) pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de aplicación o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir.

En la figura 3 se muestra el diagrama del modelo de dominio o modelo conceptual del componente para realizar el dictado automático en la elaboración de informes diagnóstico.

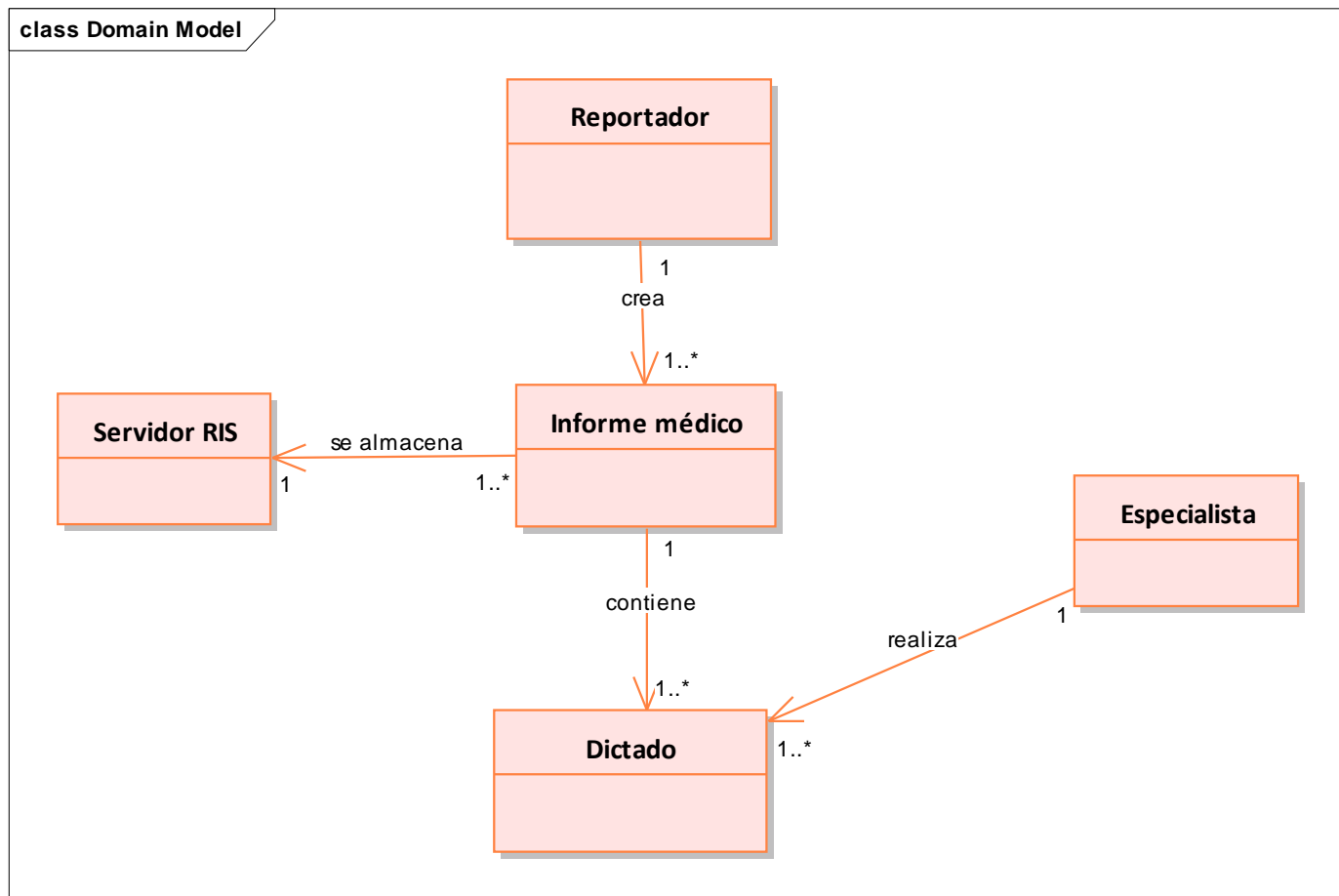


Fig. 3 Modelo de dominio.

A continuación se describen las clases conceptuales en el dominio del problema planteado:

Nombre	Reportador
Descripción	El Reportador constituye uno de los módulos del sistema PACS, tiene como objetivo principal la confección de un informe del estudio médico imagenológico realizado a un paciente para luego almacenarlo en el servidor RIS.

Tabla 1 Clase Conceptual Reportador.

Nombre	Informe médico
--------	----------------



Descripción	Es el informe que contiene el diagnóstico del estudio realizado a un paciente.
-------------	--

Tabla 2 Clase Conceptual Informe médico.

Nombre	Servidor RIS
Descripción	El servidor RIS se utiliza para almacenar los informes imagenológicos.

Tabla 3 Clase Conceptual RIS.

Nombre	Especialista
Descripción	Realiza el dictado.

Tabla 4 Clase Conceptual Usuario.

Nombre	Dictado
Descripción	Se encarga de transcribir el dictado a texto y las operaciones con los comandos.

Tabla 5 Clase Conceptual Dictado.

## 2.2. Descripción del proceso de generación de informes diagnóstico.

El proceso para la realización de informes diagnóstico, comienza con el especialista ya autenticado en el Reportador, después el mismo selecciona la opción Automatic Dictation, y se habilita la interfaz del reconocimiento de voz del sistema operativo Windows. Previamente el especialista le da un entrenamiento al sistema si aún no lo ha utilizado para tener una mayor precisión en el dictado. Luego se procede a la realización del informe, al que se le puede dar el formato que se desee mediante comandos. Por último se almacena en el servidor RIS.

## 2.3. Especificación de requisitos de software.

La especificación de requisitos de software (35) es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye los casos de uso, requerimientos funcionales y los requerimientos no funcionales, que estos describen todas las interacciones que tendrán los usuarios con la aplicación.

### 2.3.1. *Requerimientos funcionales.*

Los requerimientos funcionales (36) (RF) son capacidades o condiciones que el sistema debe cumplir. Definición de los servicios que el sistema debe proporcionar, cómo debe reaccionar y comportarse ante

situaciones particulares. Definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

La tabla 6 muestra los requisitos funcionales con una breve descripción para su mayor comprensión.

Número	Requerimiento	Descripción
RF 1.1	Iniciar el reconocimiento de voz.	Ejecuta la interfaz del reconocimiento de voz del sistema operativo Windows para comenzar el dictado.
RF 1.2	Terminar el reconocimiento de voz.	Termina el dictado automático cerrando la interfaz de reconocimiento de voz del sistema operativo Windows.
RF 1.3	Abrir diccionario.	Es un comando de voz, permite adicionar y modificar una palabra al diccionario.

Tabla 6 Requisitos Funcionales.

En la figura 4 se muestra los requisitos funcionales agrupados por paquetes.

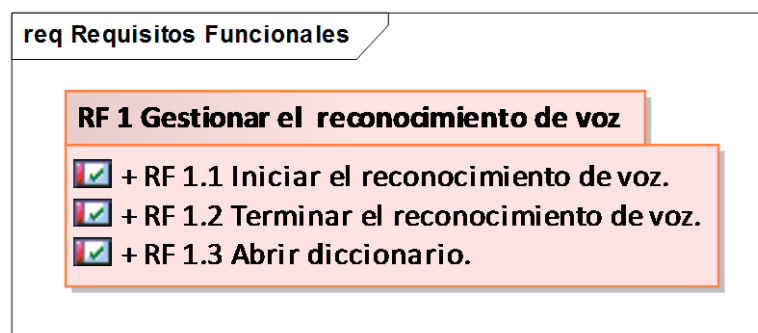


Fig. 4 Requisitos Funcionales.

### 2.3.2. Requisitos no funcionales.

Los requisitos no funcionales (36) (RNF) son propiedades o cualidades que el producto debe tener. Estos requerimientos se basan en las restricciones de los servicios o funciones ofrecidos por el sistema.

# Componente para realizar el dictado automático en la elaboración de los informes diagnóstico. **Capítulo 2**

Los requisitos no funcionales fueron nombrados asignándoles un prefijo, de acuerdo a las Pautas para la Fase de Requerimientos del Departamento SWMI. En el [Anexo 1](#) se muestra la tabla de los prefijos según la categoría del requisito.

En la tabla 11 se muestra los requisitos no funcionales con una breve descripción del sistema.

Número	Requerimiento	Descripción
RNU 1	Facilidad de empleo para usuarios con poca experiencia.	Fácil empleo para usuarios con conocimientos básicos en el manejo de aplicaciones de escritorio.
RNDI 1	Uso de <i>Framework</i> .Net 4.0.	Se especifica el uso de Microsoft <i>Framework</i> .Net 4.0 que ofrece mejoras en cuanto a administración y rendimiento. El lenguaje de programación C# depende de este <i>Framework</i> .
RNDI 2	C# como lenguaje de programación.	Se deberá utilizar C# como lenguaje de programación pues está diseñado y optimizado para la plataforma .NET.
RNDI 3	Entorno Integrado de Desarrollo Microsoft Visual Studio 2010.	Se utilizará Microsoft Visual Studio 2010 como entorno integrado de desarrollo. Este IDE utiliza como marco de trabajo el <i>Framework</i> .Net 4.0 y soporta C# como lenguaje de programación.
RNDI 4	Librerías a utilizar	Se utiliza la librería Speeplib para el reconocimiento de voz.
RNDI 5	Uso de Enterprise Architect como herramienta de modelado.	Se utilizará como herramienta de modelado Enterprise Architect en su versión 7.5.
RNDI 6	Utilizar el patrón de diseño Model-View-View-View-View-View (MVVM) para la asignación de	Utilizar el patrón de diseño Model-View-View-View-View-View (MVVM) para la asignación de

	ViewModel (MVVM).	responsabilidades.
RNDI 7	Uso de Windows Presentation Foundation para el diseño de la interfaz de usuario.	La interfaz de usuario será diseñada utilizando la tecnología Windows Presentation Foundation (WPF).
RNF 1	Disponibilidad del sistema siempre.	El sistema debe estar disponible siempre.
RNSO 1	Manual de usuario.	Manual de usuario para guiar al cliente.
RNSO 2	Expediente de proyecto actualizado.	El expediente de proyecto estará bien documentado para garantizar futuros mantenimientos.
RNE 1	Realizar el dictado de manera rápida y eficiente.	El sistema debe permitir hacer el dictado de manera rápida y eficiente.
RNFO 1	Sistema Operativo Windows 7 o superior.	Sistema Operativo Windows 7 o superior.
RNFO 2	Memoria RAM 512 MB o superior.	Memoria RAM 512 MB o superior.
RNFO 3	Procesador Pentium IV 3.0 GHz o superior.	Procesador Pentium IV 3.0 GHz o superior.
RNFO 4	Calidad de micrófono.	Para el reconocimiento de voz se necesita un micrófono con buena calidad.

Tabla 7 Requisitos no Funcionales.

En el [Anexo 2](#) se muestra el diagrama de requisitos no funcionales.

#### 2.4. Definición de los actores.

Un actor (37) es alguien o algo que interactúa con el sistema; es quien utiliza el mismo, ya es el encargado de intercambiar información con el sistema. Un actor puede ser una persona u otro sistema que se comunica con el sistema a modelar.

La tabla 8 muestra el actor que interactúa con los casos de uso del sistema.


Actor	Justificación
<div data-bbox="240 361 451 680"> <p>uc Actores</p>  <p><b>Especialista</b></p> </div>	<p>Inicia los casos de uso para realizar el dictado.</p>

Tabla 8 Actor del sistema.

### 2.5. Diagrama de casos de uso del sistema.

Los casos de uso (CU) (37)son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Cada caso de uso debe estar relacionado al menos con un actor.

La figura 5 muestra el diagrama de casos de uso del sistema.

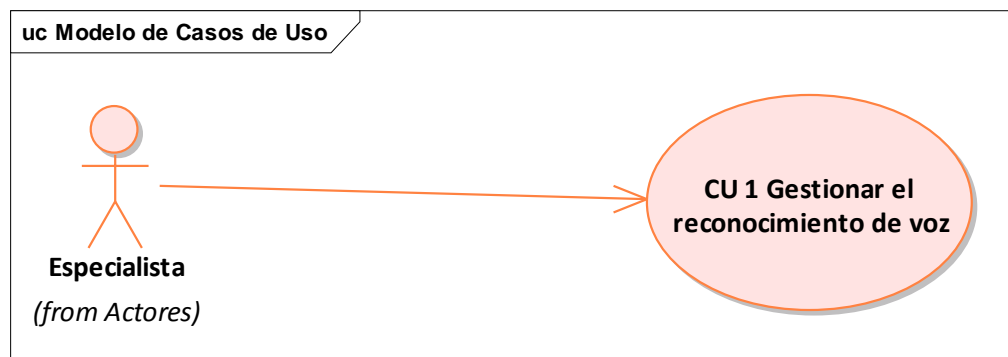


Fig. 5 Diagrama de casos de uso.

### 2.6. Descripción de los casos de uso del sistema.

La tabla 9 muestra la descripción del caso de uso del sistema.

CU Gestionar el reconocimiento de voz.

CU 1. Gestionar el reconocimiento de voz.	
<b>Objetivo</b>	El objetivo es gestionar el reconocimiento de voz.
<b>Actores</b>	Especialista (Inicia)
<b>Resumen</b>	El caso de uso comienza cuando el especialista selecciona la opción "Iniciar el reconocimiento de voz", donde el sistema permite iniciar, terminar el reconocimiento de voz y abrir el diccionario.
<b>Complejidad</b>	Media
<b>Prioridad</b>	Alta
<b>Referencias</b>	RF 1.1, RF 1.2 y RF 1.3
<b>Precondiciones</b>	El especialista se ha autenticado y ha conectado el micrófono.
<b>Postcondiciones</b>	Al concluir el dictado se logró el reconocimiento de voz.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Iniciar reconocimiento de voz &gt;</b>	
<ol style="list-style-type: none"> <li>1. El <b>especialista</b> selecciona la opción "Iniciar el reconocimiento de voz".</li> <li>2. El <b>sistema</b> muestra la interfaz de reconocimiento de voz del sistema operativo Windows y comienza el reconocimiento de voz.</li> <li>3. El <b>especialista</b> selecciona la opción "Abrir el diccionario".</li> <li>4. El <b>sistema</b> permite adicionar y modificar una palabra al diccionario.</li> <li>5. El <b>especialista</b> termina el reconocimiento de voz seleccionando la opción "Terminar el reconocimiento de voz".</li> <li>6. El <b>sistema</b> cierra la interfaz del reconocimiento de voz del sistema operativo Windows.</li> <li>7. Termina el caso de uso.</li> </ol>	

Tabla 9 CU Gestionar el reconocimiento de voz.

## 2.7. Diseño

El diseño es una etapa fundamental en la construcción de la aplicación. Durante este se desarrollan, revisan y se documentan los refinamientos progresivos de las estructuras de datos, de la estructura del programa y de los detalles procedimentales. El diseño da como resultado representaciones cuya calidad puede ser evaluada.

El diseño de la aplicación (38) es un proceso mediante el que se traducen los requisitos en una representación de la aplicación, que se acerca mucho al código fuente. La arquitectura de la aplicación y el diseño están fuertemente relacionados, en esta etapa los requerimientos son transformados en un conjunto de clases relacionadas entre sí. Además se generan artefactos que muestran cómo se realiza el proceso.

Los diagramas de clases del diseño muestran las clases participantes y las relaciones entre sí, participantes en la relación de un caso de uso. El diagrama de clases del diseño correspondiente del caso de uso arquitectónicamente significativo se muestra en el [Anexo 3](#).

Los diagramas de secuencia muestran gráficamente las interacciones del autor y de las operaciones a que dan origen. En estos se muestran la relación entre objetos para cumplir con los requerimientos planteados. En el [Anexo 4](#) se muestran el diagrama de secuencia del caso de uso.

### 2.8. Descripción de las clases

La descripción de las clases permite obtener información sobre las mismas, principalmente con las que están involucradas con los casos de uso arquitectónicamente significativos, muestra además el propósito que presenta cada una, la descripción de los atributos y métodos. Estas descripciones pueden ser consultadas en el [Anexo 5](#).

### 2.9. Modelo arquitectónico

La arquitectura de una aplicación (39) es un conjunto de decisiones importantes acerca de: la organización de un sistema, los elementos estructurales (con sus interfaces) que componen el sistema, el comportamiento del sistema especificado por las colaboraciones existentes entre los elementos estructurales, la agrupación (o separación) de elementos estructurales y del comportamiento en subsistemas progresivamente más grandes (o más pequeños) y el estilo arquitectónico que guía la organización de tales elementos, sus interfaces, sus colaboraciones y su composición.

El sistema va a utilizar como patrón arquitectónico Model View ViewModel y como patrones de diseño: inyección de dependencias y singleton; los patrones General Responsibility Assignment Software Patterns (GRASP), dentro de este el patrón controlador.

MVVM (40) es un patrón de diseño de aplicaciones para desacoplar código de interfaz de usuario y código que no sea de interfaz de usuario. Por una parte, Model, es el encargado de representar el modelo del negocio, siendo la base necesaria para la manipulación de los datos de la aplicación. En el caso de View, es la responsable de la parte visual de la aplicación, sin preocuparse en ningún momento del manejo de datos. ViewModel es la capa intermedia entre el modelo y la vista, procesando todas las peticiones que tenga la vista hacia el modelo, además de manejar las reglas del negocio, la comunicación con aplicaciones externas o consumir datos desde alguna fuente. El siguiente diagrama muestra el funcionamiento de este patrón.

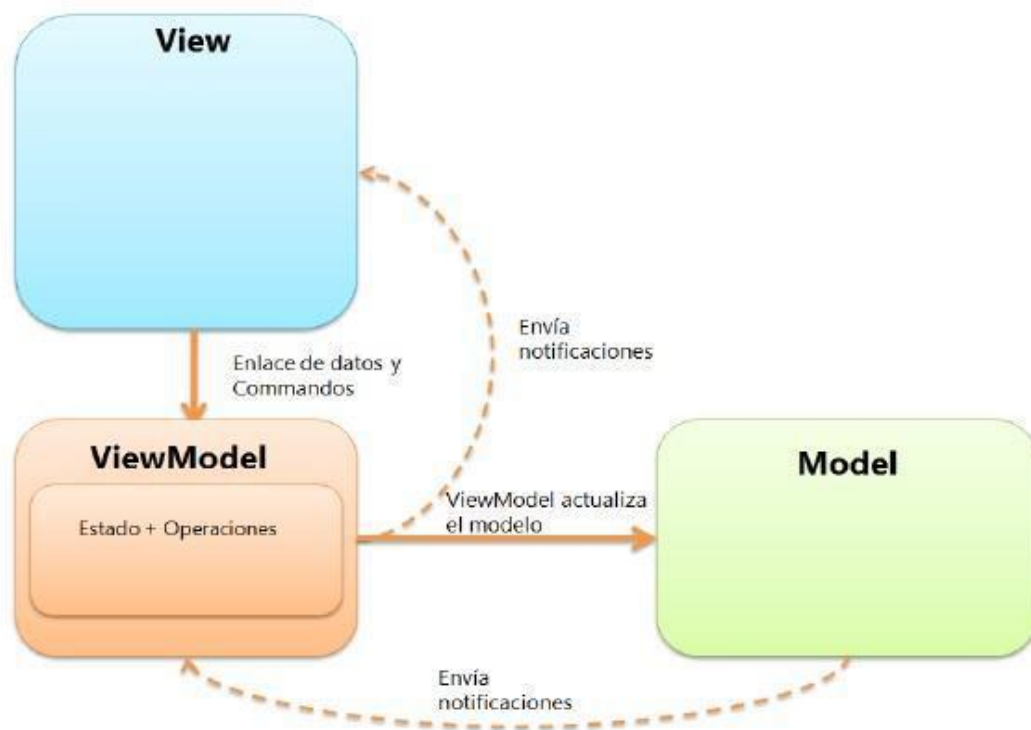


Fig. 6 Patrón Model View ViewModel.

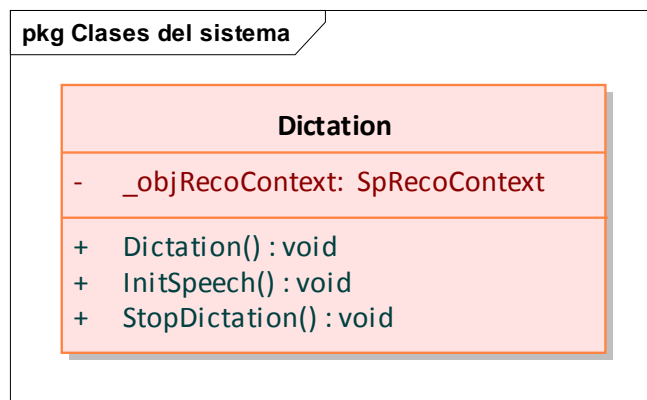
El patrón de Inyección de Dependencias (DI, por sus siglas en inglés) (41) es un caso especial de inversión de control en el que se suplen objetos/dependencias a una clase en lugar de ser la propia clase quien cree



los objetos/dependencias que necesita. El patrón Singleton que garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a ésta instancia.

Los GRASP (42), son patrones generales de la aplicación para asignación de responsabilidades, es la habilidad más importante en el análisis y diseño orientado por objetos. Respetar los principios fundamentales es uno de los factores críticos, para obtener diseños reutilizables, mantenibles y extendibles.

Patrón controlador: es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.



*Fig. 7 Clase Dictation.*

## **Consideraciones del capítulo**

Con el presente capítulo se consideró que la elaboración del diseño a través del modelo de dominio facilita el desarrollo del componente debido a sus características y contexto que se utiliza. La identificación y descripción del CU posibilitó un mejor entendimiento de los requerimientos del componente.

### CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL COMPONENTE PARA REALIZAR EL DICTADO AUTOMÁTICO DE LOS INFORMES DIAGNÓSTICO.

En el presente capítulo se muestran los artefactos que se generan en la etapa de implementación del sistema, como es el caso del diagrama de componentes que muestra la forma en que interactúan para posibilitar su correcto funcionamiento y el diagrama de despliegue que especifica la ubicación física de cada componente y mediante qué protocolos se comunicarán.

#### 3.1. Interfaz del componente.

En la figura 8 se muestra la interfaz del informe del estudio imagenológico del Reportador del CESIM que incluye el componente para el dictado automático. La misma cuenta con un campo de texto para describir la opinión médica sobre el estudio de un paciente. Esta se puede realizar mediante el teclado, o habilitando la opción “Iniciar Dictado Automático” para comenzar la confección del informe.

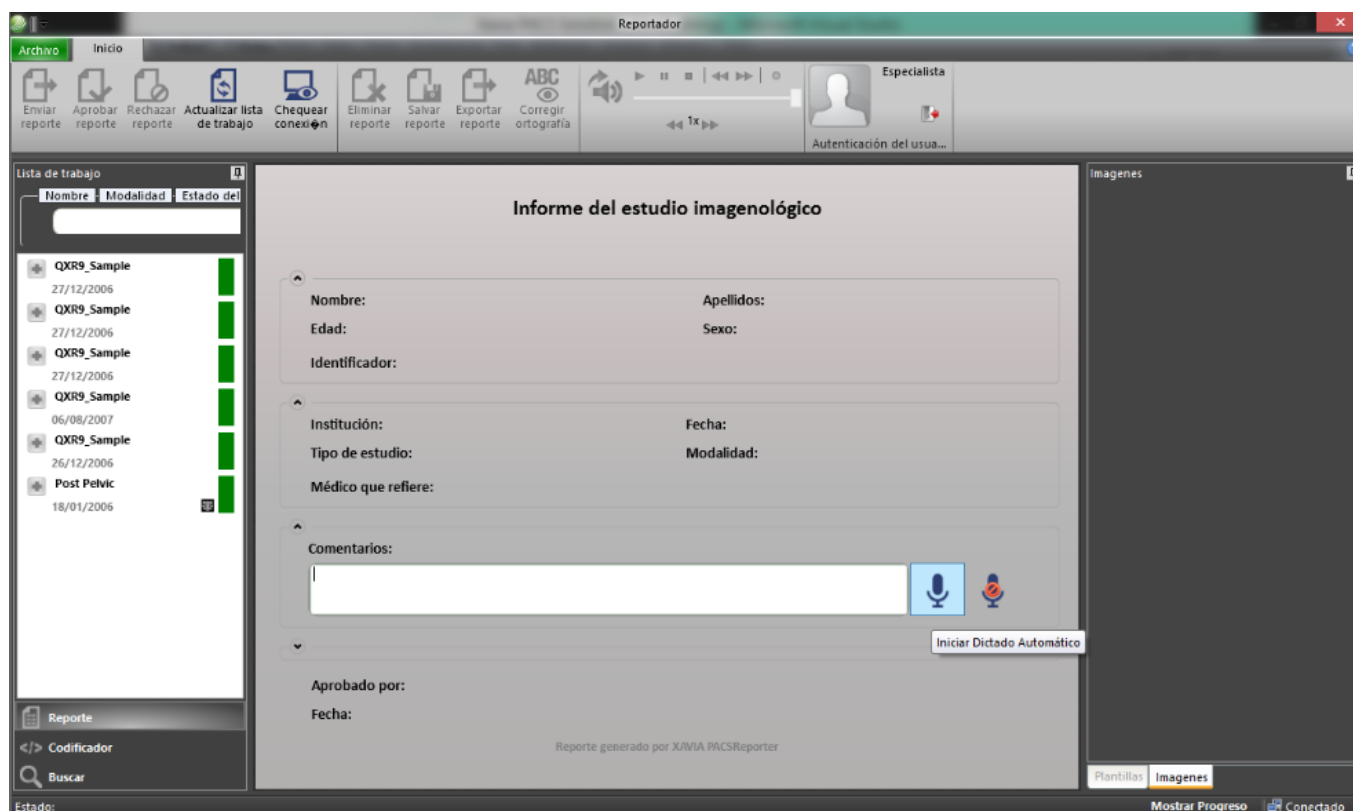


Fig. 8 Interfaz del informe del estudio imagenológico.

### 3.2. Estándar de codificación para el componente.

Los estándares de codificación (43) es un estilo de programación homogéneo en un proyecto, permite que todos los participantes lo puedan entender en menos tiempo y que el código en consecuencia sea mantenible. En la presente investigación se utilizó la convención de C# como estándar de codificación, estos le dan un aspecto coherente al código. Algunos de los elementos del estándar se describen a continuación:

#### 3.2.1. Líneas y espacios en blanco.

En la legibilidad del código se utilizan en ocasiones líneas en blanco para separar segmentos de código para los métodos, propiedades, variables y secciones lógicas dentro de los métodos. Los espacios en blanco se utilizan después de cada punto y coma, al igual que después de las comas en las funciones y declaraciones.

#### 3.2.2. Declaración de las clases.

Comenzarán con la primera letra en mayúscula. En la figura 9 se muestra un ejemplo:

```
class Dictation
```

*Fig. 9 Declaración de las clases.*

#### 3.2.3. Llaves.

Las llaves de apertura se colocarán en la línea siguiente y al mismo nivel de la línea cabecera del bloque. Las de cierre se colocarán en la línea que sigue y al mismo nivel de la última línea del bloque. Se muestra un fragmento de código donde se evidencia la utilización de las llaves en la figura 10.

```

public ICommand EndDictationCommand
{
    get
    {
        if(_endDictation == null)
        {
            _endDictation = new RelayCommand(x => Dictation.EndSpeech());
        }
        return _endDictation;
    }
}

```

Fig. 10 Utilización de las llaves.

### 3.2.4. Comentarios.

En el sistema se estableció la utilización de comentarios lineales o de una sola línea cuya sintaxis para el lenguaje C# es “//”. Se hallan encima de la línea a la que se le quiso aplicar el comentario. La figura 11 muestra se observa un fragmento de donde se utilizan los comentarios.

```

//Comando que inicia el reconocimiento de voz de Windows //
0 references
public ICommand StartDictationCommand...

```

Fig. 11 Comentarios.

### 3.2.5. Tratamiento de errores

Uno de los problemas más frecuentes en la ejecución de un sistema surge con las numerosas situaciones de error que puedan existir producto a variables que no pueden ser controladas por el sistema. Frente a situaciones como estas es necesario tomar precauciones para evitar cualquier error emergente que pueda surgir y tratarlo de la forma correcta.

Para el manejo de este tipo de situaciones C# ofrece un sistema denominado manejo de excepciones. Una excepción es una situación anormal que se da a lo largo de la ejecución del programa. El manejador de la excepción, es una estructura de control que permite ejecutar un código de acuerdo a esa situación anormal de forma controlada. En el desarrollo del componente se utilizó el manejo de excepciones en los métodos a través de bloques de estructura *try/catch*. La figura 12 muestra un ejemplo.

```
public void StartSpeech()
{
    try
    {
        _objRecoContext = new SpSharedRecoContext();
    }
    catch (Exception e)
    {
        MessageBox.Show("Conecte un micrófono!!", "Información", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
```

*Fig. 12 Tratamiento de errores.*

### 3.3. Diagrama de componentes

Un componente (29) es una parte física y reemplazable de un sistema, conformado por un conjunto de interfaces y que realiza esas interfaces. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos como ficheros ejecutables, binarios; entre otros. En un componente se puede implementar uno o varios elementos, ejemplo varias clases, sin embargo, la forma exacta en que se crea esta traza depende de cómo van a ser estructurados y modularizados los ficheros de código fuente, dado el lenguaje de programación que se esté usando.

Los diagramas de componentes (44) describen los elementos físicos del sistema y sus relaciones, representan las dependencias entre componentes de la aplicación, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. El diagrama de componente hace parte de la vista física de un sistema. Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

La figura 13 muestra el diagrama de componentes para realizar el dictado automático en la elaboración de los informes diagnóstico. El componente Reporter.dll es responsable de mostrar la información al usuario e interpretar sus acciones. Esta librería se relaciona con la librería AutomaticDictation.dll que es la responsable de configurar el reconocimiento de voz, y esta se relaciona con SpeechLib.dll que habilita la interfaz del reconocimiento de voz del sistema operativo Windows.

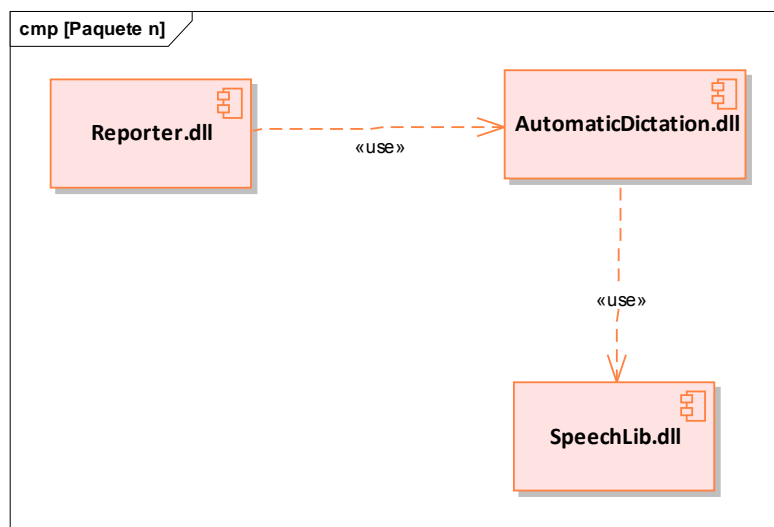


Fig. 13 Diagrama de componentes.

### 3.4. Diagrama de despliegue

Los diagramas de despliegue (45) muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Como se muestra en la figura 14 el diagrama de despliegue.

#### **Nodo Estación de reporte**

Representa las estaciones de trabajo (ordenadores) en las que va a estar instalado el Reportador y donde se realizarán actividades como la emisión de informes de estudios imagenológicos como parte del proceso de generación de un informe.

#### **Nodo PACSViewer**

Servidor para el almacenamiento, transmisión y visualización de imágenes radiológicas, documentación y reportes. Debe tener amplia capacidad debido a la gran cantidad de información que necesita archivarse y el tiempo que esta permanecerá en el mismo.

#### **Nodo Repositorio de reportes**

El repositorio de reportes es un componente del sistema alas PACS que tiene como función almacenar los informes que son enviados desde el sistema, este puede estar estandarizado (DICOM compatible) o no.

### **Nodo Servidor Worklist**

Worklist en un servidor de listas de trabajo encargado de asegurar la coherencia de la información radiológica en los servicios de imagenología de los centros hospitalarios.

### **Nodo Servidor de actualización**

Representa un repositorio de actualizaciones para los sistemas instalados.

### **Nodo Impresora**

Representa el *hardware* en el que se realizarán las actividades de impresión de reportes.

### **Nodo Micrófono**

Representa el *hardware* en el que se realizarán las actividades del reconocimiento de voz.

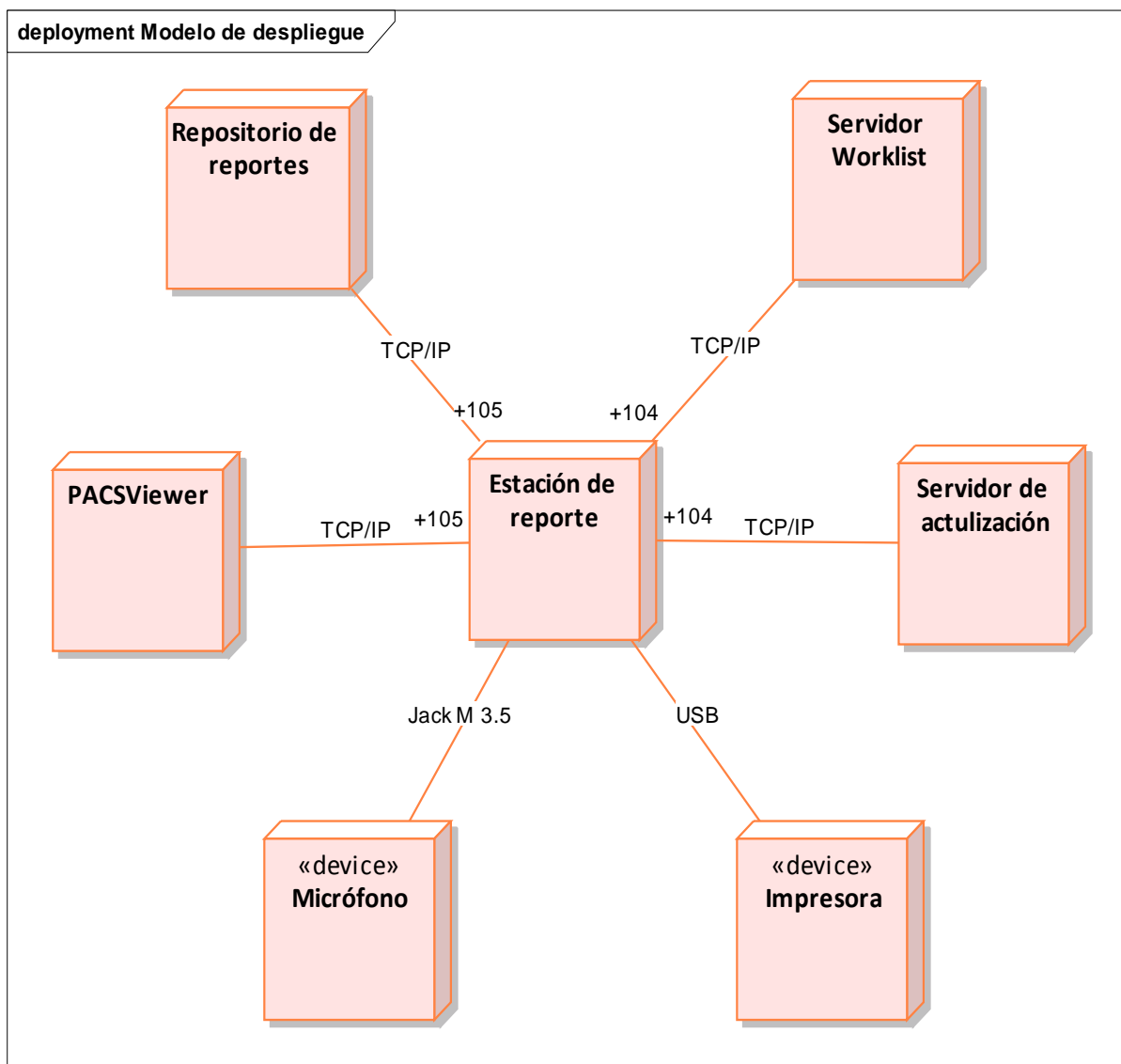


Fig. 14 Diagrama de despliegue.

### 3.5. Seguridad

La seguridad en una aplicación informática es de vital importancia, debido a que protege todos los datos manejados y almacenados por el personal encargado de efectuar las tareas de gestión de dicha información. Una de las formas de cumplir con esta tarea es mediante la ofuscación del código fuente de la aplicación.



La ofuscación (46) es una forma de proteger el código fuente de una aplicación, que permite ejecutarlo por otras personas, sin que estas puedan llegar a entender cómo funciona en realidad. Su modo de funcionamiento es representar de manera incomprensible la lógica final del código ejecutable, para dificultar o imposibilitar su ingeniería inversa. Una forma básica de realizarlo es modificando los nombres de las variables, métodos y funciones.

Para asegurar el código fuente del componente desarrollado durante la investigación, se decidió utilizar la técnica de ofuscación. En el Visual Studio se encuentra una herramienta que permite realizar esta técnica, la cual requiere una serie de pasos que se muestran a continuación:

- Ir a *tools*.
- Seleccionar *Dotfuscator Software Services*.
- Muestra la interfaz del *software* y se selecciona *Input Assemblies*.
- Se busca la dirección de la librería.
- Se selecciona *Build Project*.

### 3.6. Validación del componente desarrollado.

La validación (47) es el proceso de revisión al que se somete una aplicación para comprobar que cumple con las especificaciones. Es normalmente una parte del proceso de pruebas de la aplicación de un proyecto, que también utiliza técnicas tales como evaluaciones, inspecciones y tutoriales.

Este proceso se realiza a través de pruebas específicas. Las mismas son actividades en las cuales un sistema o uno de sus componentes se ejecuta en circunstancias previamente especificadas. Los resultados se observan y registran, y se realiza una evaluación de algún aspecto.

Actualmente existen varios tipos de pruebas (48), tal es el caso de la prueba de stress para verificar el comportamiento de una aplicación bajo una demanda excesiva. También existen las pruebas de regresión que se realizan principalmente sobre una aplicación que no se tiene documentación o ningún tipo de pruebas, por lo que estas pruebas se deben realizar antes de hacerle una modificación a la aplicación.

Otra forma de realizar pruebas es mediante el modelo V (49), que considera las pruebas como una actividad paralela al ciclo de vida del desarrollo de una aplicación y no como una actividad aislada que se realiza al final. Este tiene como objetivos minimizar los riesgos, mejorar y garantizar la calidad del sistema.

La validación del componente se basa en el modelo V como se muestra en la figura 15, del cual se describen en las siguientes secciones su modo de aplicación:

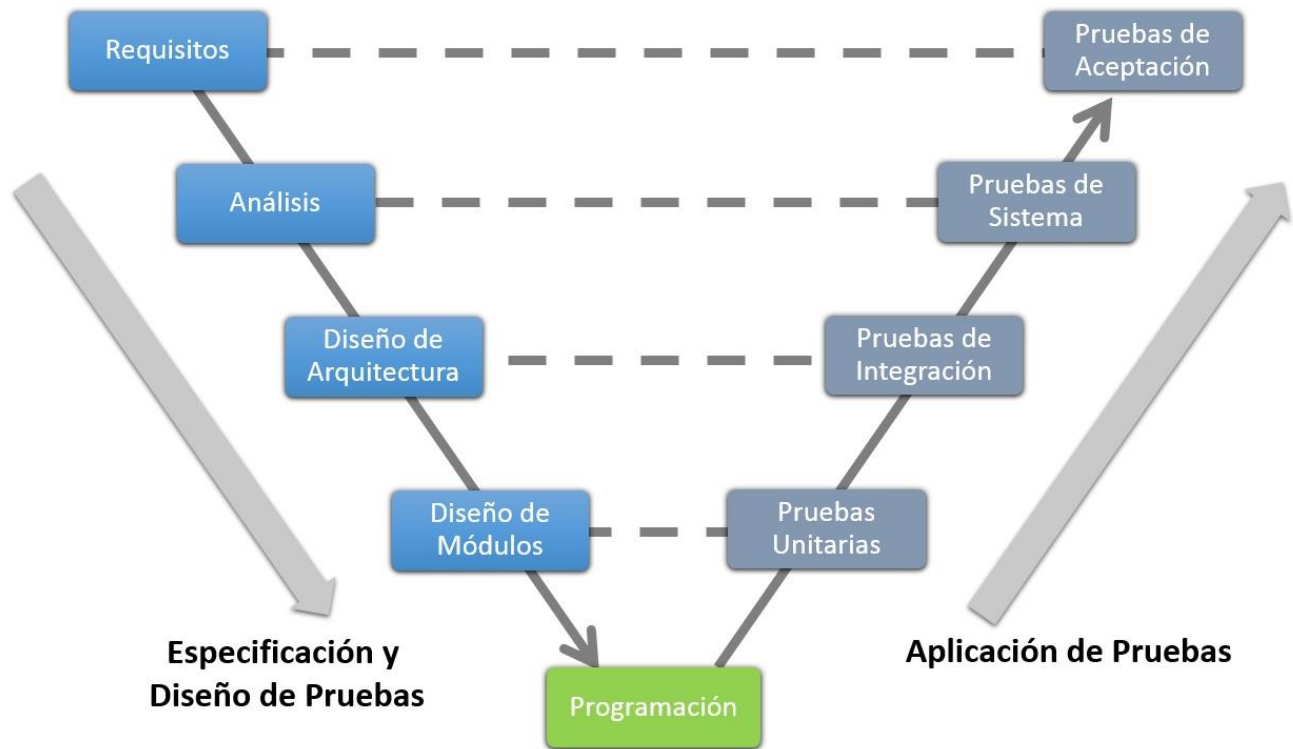


Fig. 15 Modelo V.

### 3.6.1. Pruebas unitarias.

Las pruebas unitarias buscan asegurar que el código funciona de acuerdo con las especificaciones y que el módulo lógico es válido. Durante el desarrollo del componente existieron no conformidades, tal como:

- Al invocar la librería para iniciar el reconocimiento de voz, mostró el siguiente error “*Error 1 Interop type 'SpeechLib.SpSharedRecoContextClass' cannot be embedded. Use the applicable interface instead*”, porque se estaba invocando a la clase *SpSharedRecoContextClass* cuando la invocación correcta era a la interfaz *SpSharedRecoContext*.
- Cuando se estaba insertando la imagen al botón detener, mostró el siguiente error “*Error 2 The property 'Content' is set more than once*” pues este contiene dos imágenes que se incluyen dentro de un *grid*.

La solución de estas no conformidades fueron corregidas satisfactoriamente en menos de 12 horas por el desarrollador Yasmany Reyes González.

### 3.6.2. Pruebas de integración.

Las pruebas de integración permiten identificar errores introducidos por la combinación de programas probados unitariamente. En la integración del componente al Reportador se detectaron errores tales como:

- *"Error 1 The given assembly name or codebase was invalid. (Exception from HRESULT: 0x80131047)"*.
- *"Error 24 Could not copy "obj\x86\Release\Reporter.dll" to "...\.alas PACS Binary\Reporter.dll".Exceeded retry count of 10. Failed. Reporter (Modules\Reporter) "* .
- *"Error 25 Unable to copy file "obj\x86\Release\Reporter.dll" to "...\.alas PACS Binary\Reporter.dll". The process cannot access the file '...\alas PACS Binary\Reporter.dll' because it is being used by another process. Reporter (Modules\Reporter) "*, que luego de recompilar varias veces desaparecieron.

### 3.6.3. Pruebas de sistema.

Las pruebas del sistema tienen como objetivo asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación. En el caso del componente desarrollado arrojó que de 135 palabras médicas dictadas, 32 palabras mostraron dificultad pero con posibilidad de corregir o de añadir al diccionario, para un reconocimiento de voz de un 76,3 % de precisión. En el [Anexo 6](#) se muestran las palabras comprobadas. A continuación se muestra en la figura 16 la gráfica de las palabras médicas dictadas.



Fig. 16 Gráfica de las palabras médicas dictadas.

Se realizó otra iteración de las pruebas del sistema comprobando la misma muestra antes seleccionada, y arrojó que de las 135 palabras médicas dictadas, 5 resultaron con dificultad pero con la posibilidad de corregirlas, para un reconocimiento de voz de un 96.3% de precisión. Se muestra en la figura 17 la gráfica de las palabras médicas dictadas.



Fig. 17 Gráfica de las palabras médicas dictadas.

Por lo que se considera que con un mayor entrenamiento del sistema se logra mejor precisión en el reconocimiento de voz en informes diagnóstico.

#### 3.6.4. Pruebas de aceptación

Las pruebas de aceptación determinan por parte del cliente la aprobación o no de la aplicación desarrollada. Las mismas no son realizadas en el departamento SWMI por lo mismo no se ejecutaron al componente.

## **Consideraciones del capítulo**

Se pudo definir la ubicación física de cada nodo con los que se va a comunicar la estación de trabajo donde esté siendo utilizada la aplicación y las librerías que componen dicho sistema. Además se logró implementar un componente capaz de realizar el dictado automático en la elaboración de los informes diagnóstico y se integró correctamente al Reportador del CESIM.

## **CONCLUSIONES**

A través del análisis de las herramientas de reconocimiento de voz para la creación del componente, se seleccionó SAPI 5.1 con la librería de interoperabilidad SpeechLib.dll siendo esta gratuita. La misma permite interactuar con la interfaz del reconocimiento de voz del sistema operativo Windows, lo que facilita la creación de comandos y la conversión de voz a texto.

El empleo del patrón arquitectónico MVVM permitió que la interfaz de usuario y la lógica del negocio se implementaran de forma independiente.

El componente desarrollado brinda a los especialistas una herramienta que mejora la calidad, agilidad y entrega en menor tiempo de los informes diagnóstico.

Los resultados obtenidos durante la validación del componente para el dictado automático y la integración al Reportador del CESIM correctamente, arrojó un nivel de precisión de 76.3% en el reconocimiento de voz.

---

## **RECOMENDACIONES**

- Realizar una librería propia del Reportador del CESIM para el reconocimiento de voz que no dependa del sistema operativo.

## REFERENCIAS BIBLIOGRÁFICAS

1. González, Victoria Ramos. Coit. *Las TIC en el sector de la salud*. [En línea] Junio- Julio de 2007. [Citado el: 9 de octubre de 2013.] <http://www.coit.es/publicaciones/bit/bit163/41-45.pdf>.
2. Definicion.de imagenología. *Definicion.de imagenología*. [En línea] 2008. [Citado el: 9 de octubre de 2013.] <http://definicion.de/imagenologia/>.
3. EL BLOG DE ACTUALMED. actualmed. *¿Qué es un servidor PACS? ¿Por qué necesito uno?* [En línea] 20 de octubre de 2010. [Citado el: 11 de octubre de 2013.] <http://www.actualmed.com/blog/2010/10/20/servidor-pacs-dicom-server/>.
4. Izaguirre, Leodan Vega. *Alas RIS Sistema de gestión de información radiológica*. La Habana : Universidad de Ciencias Informáticas., 2008. Tesis.
5. Marquina, Agustín Álvarez. *Historia de los sistemas de reconocimiento automático del habla*. Madrid : Universidad Plitécnica de Madrid, 2001.
6. Fernández, Amparo Varona. *Antecedentes y desarrollo de los sistemas actuales de reconocimiento automático del habla*. s.l. : Univ. del País Vasco, 1997.
7. RABINER, L.R., JUANG, B.H. *Fundamental of speech recognition*. New Jersey : PTR Prentice-Hall, 1993.
8. Juang, B. H. *The Past, Present, and Future of Speech Processing*. s.l. : IEEE Signal Processing Magazine, 1998.
9. Bello, Richard Nolberto Rojas. *Diseño y desarrollo de prototipo de sistema de traducción instantanea de habla y transmisión en tiempo real, sobre el protocolo RTP utilizando tecnologías de reconocimiento de voz*. Chile : Universidad Austral de Chile, 2005.
10. Martínez, Alfonso Martínez. *Diseño y construcción de una arquitectura de línea de producto para sistemas PACS*. [En línea] 2001. [Citado el: 5 de Noviembre de 2013.] <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r43242.PDF>.
11. Pardo, D. Garcías González y J. Socías. *Arquitectura para el sistema de almacenamiento de informes de estudios imagenológicos*. La Habana : Tesis, 2010.



12. Daniel García González, Jacqueline Socias Pardo. *Arquitectura para el sistema de almacenamiento de informes de estudios imagenológicos*. La Habana : Universidad de las Ciencias Informáticas, 2010.
13. Ing. Maryelines Labrada Madrigal, Ing. Arellys Rivero Castro, Arturo Yasmany González Yera. *Sistema para la edición de informes de estudios imagenológicos con gestión dinámicas de interfaces de usuario. Informática en salud 2013*. [En línea] 2013. [Citado el: 20 de enero de 2014.] <http://www.informatica2013.sld.cu/index.php/informaticasalud/2013/paper/viewFile/444/259>. ISBN: 978-959-7213-02-4.
14. Nuance. nuance.es. *Guía del Usuario*. [En línea] [Citado el: 22 de Febrero de 2014.] [http://www.nuance.es/ucmprod/groups/dragon/@web-es/documents/collateral/nq\\_020444.pdf](http://www.nuance.es/ucmprod/groups/dragon/@web-es/documents/collateral/nq_020444.pdf).
15. Arturo Yasmany González Yera, Yusleidis de la Caridad Cepero González. *Sistema para la edición de informes de estudios imagenológicos con gestión dinámica de interfaces de usuario*. La Habana : Universidad de las Ciencias Informáticas, 2012. 005.368-Cep-S.
16. Philips. Soluciones profesionales de dictado y grabadoras de dictado. *Soluciones profesionales de dictado y grabadoras de dictado*. [En línea] 2013. [Citado el: 22 de Febrero de 2014.] [https://www.dictation.philips.com/ce/como-hacemos-la-diferencia/news/el\\_reconocimiento\\_de\\_voz\\_y\\_el\\_flujo\\_de\\_trabajo/](https://www.dictation.philips.com/ce/como-hacemos-la-diferencia/news/el_reconocimiento_de_voz_y_el_flujo_de_trabajo/).
17. Rojo, Marcial García. Reconocimiento de voz en Medicina. *Reconocimiento de voz en Medicina*. [En línea] [Citado el: 23 de Febrero de 2014.] <http://www.uninet.edu/conganat/IIICVHAP/conferencias/006/voz.htm#soluciones>.
18. Yunier Leyva Rivera, Roaldys García Robaina. *Herramienta para el manejo de flujos de trabajos y su integración en una arquitectura empresarial de JEE*. La Habana : Universidad de las Ciencias Informáticas, 2007. 005-12-Ley-H-TD-1372-08.
19. Oracle Corporation. Herramientas de Desarrollo | Oracle. *Oracle*. [En línea] Oracle Corporation, 2014. [Citado el: 30 de Marzo de 2014.] <http://www.oracle.com/lad/products/middleware/tools/index.html?ssSourceSiteId=ocomen>.
20. Microsoft. Información general y conceptual sobre .NET Framework. *MSDN*. [En línea] [Citado el: 15 de Diciembre de 2013.] <http://msdn.microsoft.com/es-ve/library/zw4w595w.aspx>.

21. Sells, Chris y Griffiths, Ian. *Programming WPF*. s.l. : O'Reilly Media, 2007. ISBN-10: 0-596-51037-3/ISBN-13: 978-0-596-51037-4.
22. Mora, David. Patrón Modelo-Vista-Modelo de Vista (MVVM). *Scribd*. [En línea] 28 de Mayo de 2010. [Citado el: 15 de Noviembre de 2013.] <http://maromasdigitales.net/2010/05/patron-mvvm-explicado/>.
23. Sharp, John. *Microsoft Press Microsoft Visual C Sharp 2010. Step by Step*. Washington : Microsoft Press, 2010.
24. Microsoft. Visual Studio 2010. *Microsoft*. [En línea] 2010. [Citado el: 15 de Noviembre de 2013.] <http://www.microsoft.com/spain/visualstudio/products/2010-editions/ultimate>.
25. Oracle Corporation. Java Speech API Frequently Asked Questions. *Oracle*. [En línea] Oracle Corporation, 9 de Noviembre de 2013. [Citado el: 20 de Noviembre de 2014.] <http://www.oracle.com/technetwork/java/jsapifaq-135248.html>.
26. Microsoft SAPI. Microsoft SAPI 4.0a. *Microsoft*. [En línea] Microsoft, 2013. [Citado el: 16 de Noviembre de 2013.] <http://www.microsoft.com/speech/download/old/sdk40a.asp>.
27. Cuesta, Adrian Serafín Lamadrid. *Herramienta para la detección de vulnerabilidades en el código fuente del repositorio de GNU/Linux Nova*. Ciudad de La Habana : Universidad de las Ciencias Informáticas, 2012. 005.12-Lam-H.
28. Sparx Systems. *Enterprise Architect User Guide*. 2009.
29. Yania Junco López, Ariel Felipe Camacho Díaz. *Desarrollo del módulo de columna para el sistema de Planificación Quirúrgica Ortopédica*. La Habana : Universidad de las Ciencias Informáticas, 2012.
30. Patricio Salinas Caro, Nancy Histchfeld K. Tutorial de UML. [En línea] [Citado el: 25 de Enero de 2014.] <http://users.dcc.uchile.cl/~psalinas/uml/introduccion.html>.
31. Introducción a UML. [En línea] [Citado el: 25 de Enero de 2014.] <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-basics.html>.
32. GUILLERMO SOLARTE, LUIS MUÑOZ, BIBIANA ARIAS. MODELOS DE CALIDAD PARA PROCESOS DE SOFTWARE. *Scientia Et Technica*. [En línea] Agosto de 2009. [Citado el: 13 de marzo de 2014.] <http://www.redalyc.org/pdf/849/84916714069.pdf>. 0122-1701.

33. wordpress. *Tecnología y Synergix*. [En línea] 10 de Julio de 2008. [Citado el: 20 de Marzo de 2014.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
34. Larman, Craig. *MODELO DEL DOMINIO*. [En línea] 2003. [Citado el: 3 de abril de 2014.] <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>.
35. JUNTA DE ANDALUCÍA. Especificación de Requisitos del Sistema. *Marco de Desarrollo de la Junta de Andalucía*. [En línea] 2010. <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/407>.
36. Capítulo 3 DEFINICIÓN REQUERIMIENTOS. *Metodología Gestión de Requerimientos*. [En línea] Google Sites. [Citado el: 20 de Febrero de 2014.] <https://sites.google.com/site/metodologiareq/capitulo-iii>.
37. Ceria, Santiago. Casos de Uso – Un Método Práctico para Explorar Requerimientos. [En línea] Febrero de 2001. [Citado el: 20 de Febrero de 2014.] [http://www-2.dc.uba.ar/materias/isoft1/2001\\_2/apuntes/CasosDeUso.pdf](http://www-2.dc.uba.ar/materias/isoft1/2001_2/apuntes/CasosDeUso.pdf).
38. cmapspublic. *Fundamentos del diseño de software*. [En línea] [Citado el: 27 de Marzo de 2014.] <http://cmapspublic.ihmc.us/rid=1LQ5KGY9H-J0CYGK-3GBN/FundamentosDiseno.pdf>.
39. Kruchten, Philippe. *The Rational Unified Process*. 1998.
40. Windows. Usar el patrón Model-View-ViewModel (MVVM). [En línea] <http://msdn.microsoft.com/es-es/library/windows/apps/jj883732.aspx>.
41. Microsoft| Architecture. *Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0*. 2010.
42. Introducción a los patrones de Software. [En línea] <http://sistemas.uniandes.edu.co/~isis2701/dokuwiki/lib/exe/fetch.php?media=isis2701-patronesgrasp.pdf>.
43. Calleja, Manuel Arias. *Carmen. Estándares de codificación*. Ciudad de La Habana : Universidad de Ciencias Informáticas.
44. D.S., Wilder. diagramacomponente. *Diagrama de Componentes*. [En línea] mayo de 2009. [Citado el: 3 de abril de 2014.] <http://diagramacomponente.blogspot.com/>.
45. Marca Huallpara Hugo Michael, Quisbert Limachi Nancy Susana. *Diagrama de despliegue.doc*. [En línea] [Citado el: 3 de abril de 2014.] [virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc](http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc).

46. PreEmptive Solutions. Objetivo de la ofuscación. *Developer Network*. [En línea] Microsoft, 2002. [Citado el: 25 de Mayo de 2014.] [http://msdn.microsoft.com/es-es/library/ms227296\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/ms227296(v=vs.90).aspx).
47. Ceniceros, Juan Carlos Espinosa. La Verificación y Validación de Software. *Blog Juan Carlos Espinosa Ceniceros*. [En línea] 7 de Agosto de 2012. [Citado el: 15 de Marzo de 2014.] <http://juankenny.blogspot.com/2012/08/vvs-la-verificacion-y-validacion-de.html>.
48. Montoya, Eliu. Tipo de Pruebas para Desarrollo de Software. *Java & Arch Blog*. [En línea] 11 de Septiembre de 2012. [Citado el: 15 de Marzo de 2014.] <http://javablog.eliumontoya.com/home/tipodepruebasparadesarrollodesoftware#TOC-Pruebas-de-Stress>.
49. Service Desk Insitute. El modelo V sus ventajas y desventajas. *sdila.com/*. [En línea] 2010. [Citado el: 25 de Marzo de 2014.] [http://www.sdila.com/index2.php?option=com\\_docman&task=doc\\_view&gid=47&Itemid=227](http://www.sdila.com/index2.php?option=com_docman&task=doc_view&gid=47&Itemid=227).

## WEBGRAFÍA

González, Victoria Ramos. Coit. *Las TIC en el sector de la salud*. [En línea] Junio- Julio de 2007. [Citado el: 9 de octubre de 2013.] <http://www.coit.es/publicaciones/bit/bit163/41-45.pdf>.

Definicion.de imagenología. *Definicion.de imagenología*. [En línea] 2008. [Citado el: 9 de octubre de 2013.] <http://definicion.de/imagenologia/>.

EL BLOG DE ACTUALMED. actualmed. *¿Qué es un servidor PACS? ¿Por qué necesito uno?* [En línea] 20 de octubre de 2010. [Citado el: 11 de octubre de 2013.] <http://www.actualmed.com/blog/2010/10/20/servidor-pacs-dicom-server/>.

Martínez, Alfonso Martínez. *Diseño y construcción de una arquitectura de línea de producto para sistemas PACS*. [En línea] 2001. [Citado el: 5 de Noviembre de 2013.] <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r43242.PDF>.

Ing. Maryelines Labrada Madrigal, Ing. Arelys Rivero Castro, Arturo Yasmany González Yera. *Sistema para la edición de informes de estudios imagenológicos con gestión dinámicas de interfaces de usuario*. Informática en salud 2013. [En línea] 2013. [Citado el: 20 de enero de 2014.] <http://www.informatica2013.sld.cu/index.php/informaticasalud/2013/paper/viewFile/444/259>. ISBN: 978-959-7213-02-4.

Nuance. nuance.es. *Guía del Usuario*. [En línea] [Citado el: 22 de Febrero de 2014.] [http://www.nuance.es/ucmprod/groups/dragon/@web-es/documents/collateral/nq\\_020444.pdf](http://www.nuance.es/ucmprod/groups/dragon/@web-es/documents/collateral/nq_020444.pdf).

Philips. *Soluciones profesionales de dictado y grabadoras de dictado*. Soluciones profesionales de dictado y grabadoras de dictado. [En línea] 2013. [Citado el: 22 de Febrero de 2014.] [https://www.dictation.philips.com/ce/como-hacemos-la-diferencia/news/el\\_reconocimiento\\_de\\_voz\\_y\\_el\\_flujo\\_de\\_trabajo/](https://www.dictation.philips.com/ce/como-hacemos-la-diferencia/news/el_reconocimiento_de_voz_y_el_flujo_de_trabajo/).

Rojo, Marcial García. *Reconocimiento de voz en Medicina*. Reconocimiento de voz en Medicina. [En línea] [Citado el: 23 de Febrero de 2014.] <http://www.uninet.edu/conganat/IIICVHAP/conferencias/006/voz.htm#soluciones>.

- Oracle Corporation. *Herramientas de Desarrollo | Oracle. Oracle*. [En línea] Oracle Corporation, 2014. [Citado el: 30 de Marzo de 2014.] <http://www.oracle.com/lad/products/middleware/tools/index.html?ssSourceSiteId=ocomen>.
- Microsoft. *Información general y conceptual sobre .NET Framework. MSDN*. [En línea] [Citado el: 15 de Diciembre de 2013.] <http://msdn.microsoft.com/es-ve/library/zw4w595w.aspx>.
- Mora, David. *Patrón Modelo-Vista-Modelo de Vista (MVVM)*. Scribd. [En línea] 28 de Mayo de 2010. [Citado el: 15 de Noviembre de 2013.] <http://maromasdigitales.net/2010/05/patron-mvvm-explicado/>.
- Microsoft. *Visual Studio 2010*. Microsoft. [En línea] 2010. [Citado el: 15 de Noviembre de 2013.] <http://www.microsoft.com/spain/visualstudio/products/2010-editions/ultimate>.
- Oracle Corporation. *Java Speech API Frequently Asked Questions*. Oracle. [En línea] Oracle Corporation, 9 de Noviembre de 2013. [Citado el: 20 de Noviembre de 2014.] <http://www.oracle.com/technetwork/java/jsapifaq-135248.html>.
- Microsoft SAPI. *Microsoft SAPI 4.0a*. Microsoft. [En línea] Microsoft, 2013. [Citado el: 16 de Noviembre de 2013.] <http://www.microsoft.com/speech/download/old/sdk40a.asp>.
- Salinas Caro Patricio, Nancy Histchfeld K. *Tutorial de UML*. [En línea] [Citado el: 25 de Enero de 2014.] <http://users.dcc.uchile.cl/~psalinas/uml/introduccion.html>.
- Introducción a UML*. [En línea] [Citado el: 25 de Enero de 2014.] <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-basics.html>.
- GUILLERMO SOLARTE, LUIS MUÑOZ, BIBIANA ARIAS. *MODELOS DE CALIDAD PARA PROCESOS DE SOFTWARE*. Scientia Et Technica. [En línea] Agosto de 2009. [Citado el: 13 de marzo de 2014.] <http://www.redalyc.org/pdf/849/84916714069.pdf>. 0122-1701.
- wordpress. *Tecnología y Synergix*. [En línea] 10 de Julio de 2008. [Citado el: 20 de Marzo de 2014.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
- Larman, Craig. *MODELO DEL DOMINIO*. [En línea] 2003. [Citado el: 3 de abril de 2014.] <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>.
- JUNTA DE ANDALUCÍA. *Especificación de Requisitos del Sistema*. Marco de Desarrollo de la Junta de Andalucía. [En línea] 2010. <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/407>.

# Componente para realizar el dictado automático en la elaboración de los informes diagnóstico. **Bibliografía**

Capítulo 3 DEFINICIÓN REQUERIMIENTOS. *Metodología Gestión de Requerimientos*. [En línea] Google Sites. [Citado el: 20 de Febrero de 2014.] <https://sites.google.com/site/metodologiareq/capitulo-iii>.

Ceria, Santiago. Casos de Uso – *Un Método Práctico para Explorar Requerimientos*. [En línea] Febrero de 2001. [Citado el: 20 de Febrero de 2014.] [http://www-2.dc.uba.ar/materias/isoft1/2001\\_2/apuntes/CasosDeUso.pdf](http://www-2.dc.uba.ar/materias/isoft1/2001_2/apuntes/CasosDeUso.pdf).

cmappublic. *Fundamentos del diseño de software*. [En línea] [Citado el: 27 de Marzo de 2014.] <http://cmappublic.ihmc.us/rid=1LQ5KGY9H-J0CYGK-3GBN/FundamentosDiseno.pdf>.

Windows. *Usar el patrón Model-View-ViewModel (MVVM)*. [En línea] <http://msdn.microsoft.com/es-es/library/windows/apps/jj883732.aspx>.

*Introducción a los patrones de Software*. [En línea] <http://sistemas.uniandes.edu.co/~isis2701/dokuwiki/lib/exe/fetch.php?media=isis2701-patronesgrasp.pdf>.

D.S., Wilder. diagramacomponente. *Diagrama de Componentes*. [En línea] mayo de 2009. [Citado el: 3 de abril de 2014.] <http://diagramacomponente.blogspot.com/>.

Marca Huallpara Hugo Michael, Quisbert Limachi Nancy Susana. *Diagrama de despliegue.doc*. [En línea] [Citado el: 3 de abril de 2014.] [virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc](http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc).

Ceniceros, Juan Carlos Espinosa. Blog Juan Carlos Espinosa Ceniceros. *La Verificación y Validación de Software*. [En línea] 7 de Agosto de 2012. [Citado el: 15 de Marzo de 2014.] <http://juankenny.blogspot.com/2012/08/vvs-la-verificacion-y-validacion-de.html>.

Montoya, Eliu. Java & Arch Blog. *Tipo de Pruebas para Desarrollo de Software*. [En línea] 11 de Septiembre de 2012. [Citado el: 15 de Marzo de 2014.] <http://javablog.eliumontoya.com/home/tipodepruebasparadesarrollodesoftware#TOC-Pruebas-de-Stress>.

Service Desk Insitute. *El modelo V sus ventajas y desventajas*. *sdila.com/*. [En línea] 2010. [Citado el: 25 de Marzo de 2014.] [http://www.sdila.com/index2.php?option=com\\_docman&task=doc\\_view&gid=47&Itemid=227](http://www.sdila.com/index2.php?option=com_docman&task=doc_view&gid=47&Itemid=227).

Herman, S. J. Accuracy of a voice-to-text personal dictation system in the generation of radiology reports. *Diagnostic Imaging and Related Sciences*. [En línea] American Journal of Roentgenology , Julio de 1995. <http://www.ajronline.org/doi/abs/10.2214/ajr.165.1.7785581>.

AGFA Health Care. *AGFA Health Care*. [En línea] [Citado el: 05 de mayo de 2014.] [http://www.agfahealthcare.com/latam/es/main/products\\_services/ris\\_pacs\\_reporting/pacs/impax\\_60.jsp](http://www.agfahealthcare.com/latam/es/main/products_services/ris_pacs_reporting/pacs/impax_60.jsp).

PreEmptive Solutions. Developer Network. *Objetivo de la ofuscación*. [En línea] Microsoft, 2002. [Citado el: 25 de Mayo de 2014.] [http://msdn.microsoft.com/es-es/library/ms227296\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/ms227296(v=vs.90).aspx).

## **BIBLIOGRAFÍA**

Izaguirre, Leodan Vega. *Alas RIS Sistema de gestión de información radiológica*. La Habana : Universidad de Ciencias Informáticas., 2008. Tesis.

Marquina, Agustín Álvarez. *Historia de los sistemas de reconocimiento automático del habla*. Madrid : Universidad Politécnica de Madrid, 2001.

Fernández, Amparo Varona. *Antecedentes y desarrollo de los sistemas actuales de reconocimiento automático del habla*. s.l. : Univ. del País Vasco, 1997.

RABINER, L.R., JUANG, B.H. *Fundamental of speech recognition*. New Jersey : PTR Prentice-Hall, 1993.

Juang, B. H. *The Past, Present, and Future of Speech Processing*. s.l. : IEEE Signal Processing Magazine, 1998.

Bello, Richard Nolberto Rojas. *Diseño y desarrollo de prototipo de sistema de traducción instantanea de habla y transmisión en tiempo real, sobre el protocolo RTP utilizando tecnologías de reconocimiento de voz*. Chile : Universidad Austral de Chile, 2005.

Pardo, D. Garcías González y J. Socias. *Arquitectura para el sistema de almacenamiento de informes de estudios imagenológicos*. La Habana : Tesis, 2010.

Daniel García González, Jacqueline Socias Pardo. *Arquitectura para el sistema de almacenamiento de informes de estudios imagenológicos*. La Habana : Universidad de las Ciencias Informáticas, 2010.

Arturo Yasmany González Yera, Yusleidis de la Caridad Cepero González. *Sistema para la edición de informes de estudios imagenológicos con gestión dinámica de interfaces de usuario*. La Habana : Universidad de las Ciencias Informáticas, 2012. 005.368-Cep-S.



- Yunier Leyva Rivera, Roaldys García Robaina. *Herramienta para el manejo de flujos de trabajos y su integración en una arquitectura empresarial de JEE*. La Habana : Universidad de las Ciencias Informáticas, 2007. 005-12-Ley-H-TD-1372-08.
- Sells, Chris y Griffiths, Ian. *Programming WPF*. s.l. : O'Reilly Media, 2007. ISBN-10: 0-596-51037-3/ISBN-13: 978-0-596-51037-4.
- Sharp, John. *Microsoft Press Microsoft Visual C Sharp 2010. Step by Step*. Washington : Microsoft Press, 2010.
- Cuesta, Adrian Serafín Lamadrid. *Herramienta para la detección de vulnerabilidades en el código fuente del repositorio de GNU/Linux Nova*. Ciudad de La Habana : Universidad de las Ciencias Informáticas, 2012. 005.12-Lam-H.
- Sparx Systems. *Enterprise Architect User Guide*. 2009.
- Yania Junco López, Ariel Felipe Camacho Díaz. *Desarrollo del módulo de columna para el sistema de Planificación Quirúrgica Ortopédica*. La Habana : Universidad de las Ciencias Informáticas, 2012.
- Kruchten, Philippe. *The Rational Unified Process*. 1998.
- Microsoft| Architecture. *Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0*. 2010.
- Calleja, Manuel Arias. *Carmen. Estándares de codificación*. Ciudad de La Habana : Universidad de Ciencias Informáticas.
- Llorente, César de la Torre. *Guía de Arquitectura N-capas orientada al dominio con .NET 4.0*. 2010.
- Astudillo, Marcello Visconti y Hernán. *Fundamentos de Ingeniería de Software*.

## ANEXOS

Anexo 1. Prefijos de los requerimientos no funcionales por categoría.

Categoría	Prefijo	Ejemplo
Usabilidad	RNU	RNU 1. Requisito de usabilidad A RNU 2. Requisito de usabilidad B
Diseño e Implementación	RNDI	RNDI 1. Requisito de diseño e implementación
Legal	RNL	RNL 1. Requisito legal
Fiabilidad	RNF	RNF 1. Requisito de fiabilidad
Seguridad	RNS	RNS 1. Requisito de seguridad
Eficiencia	RNE	RNE 1. Requisito de eficiencia
Funcionamiento	RNFO	RNFO 1. Requisito de funcionamiento
Soporte	RNSO	RNSO 1. Requisito de soporte
Interfaz de usuario	RNIU	RNIU 1. Requisito de interfaz de usuario
Interconexión	RNI	RNI 1. Requisito de interconexión
Componentes Comprados	RNCC	RNCC 1. Requisito de componentes comprados
Licencia	RNLI	RNLI 1. Requisito de licencia

## Anexo 2. Requisitos Funcionales.

### pkg Requerimientos No Funcionales

#### Diseño e Implementación

- + RNDI 1. Uso de Framework.Net 4.0
- + RNDI 2. C# como lenguaje de programación.
- + RNDI 3. Entorno Integrado de Desarrollo Microsoft Visual Studio 2010.
- + RNDI 4. Librerías a utilizar
- + RNDI 5. Uso de Enterprise Architect como herramienta de modelado.
- + RNDI 6. Utilizar el patrón de diseño Model-View-ViewModel (MVVM).
- + RNDI 7. Uso de Windows Presentation Foundation para el diseño de la interfaz de usuario.

#### Eficiencia

- + RNE 1. Realizar el dictado de manera rápida y eficiente

#### Soporte

- + RNSO 1. Manual de usuario.
- + RNSO 2. Expediente de proyecto actualizado.

#### Fiabilidad

- + RNF 1. Disponibilidad del sistema siempre.

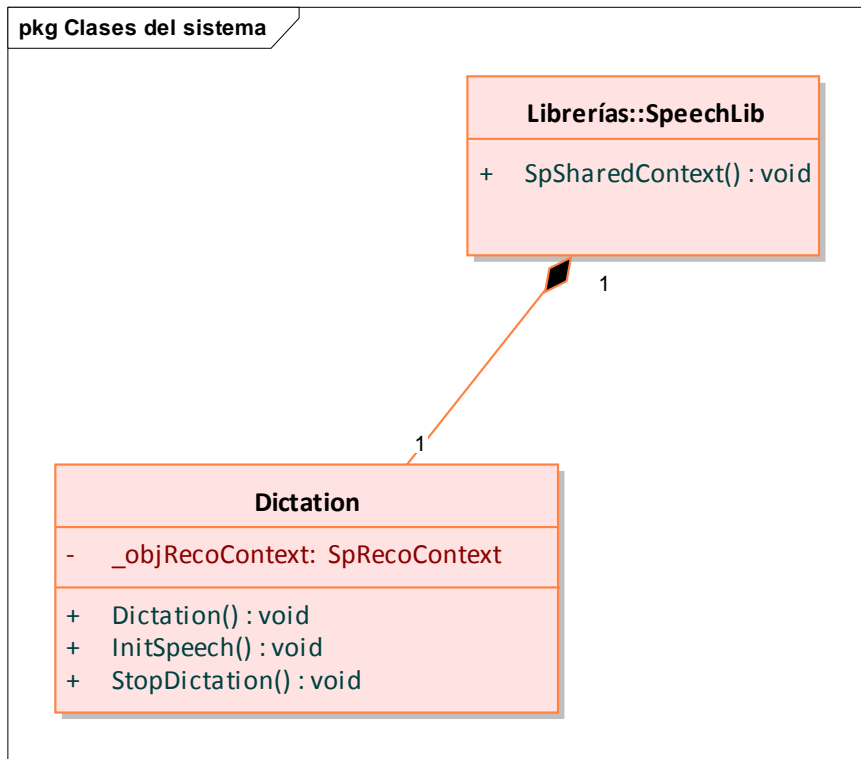
#### Usabilidad

- + RNU 1. Facilidad de empleo para usuarios con poca experiencia.

#### Funcionamiento

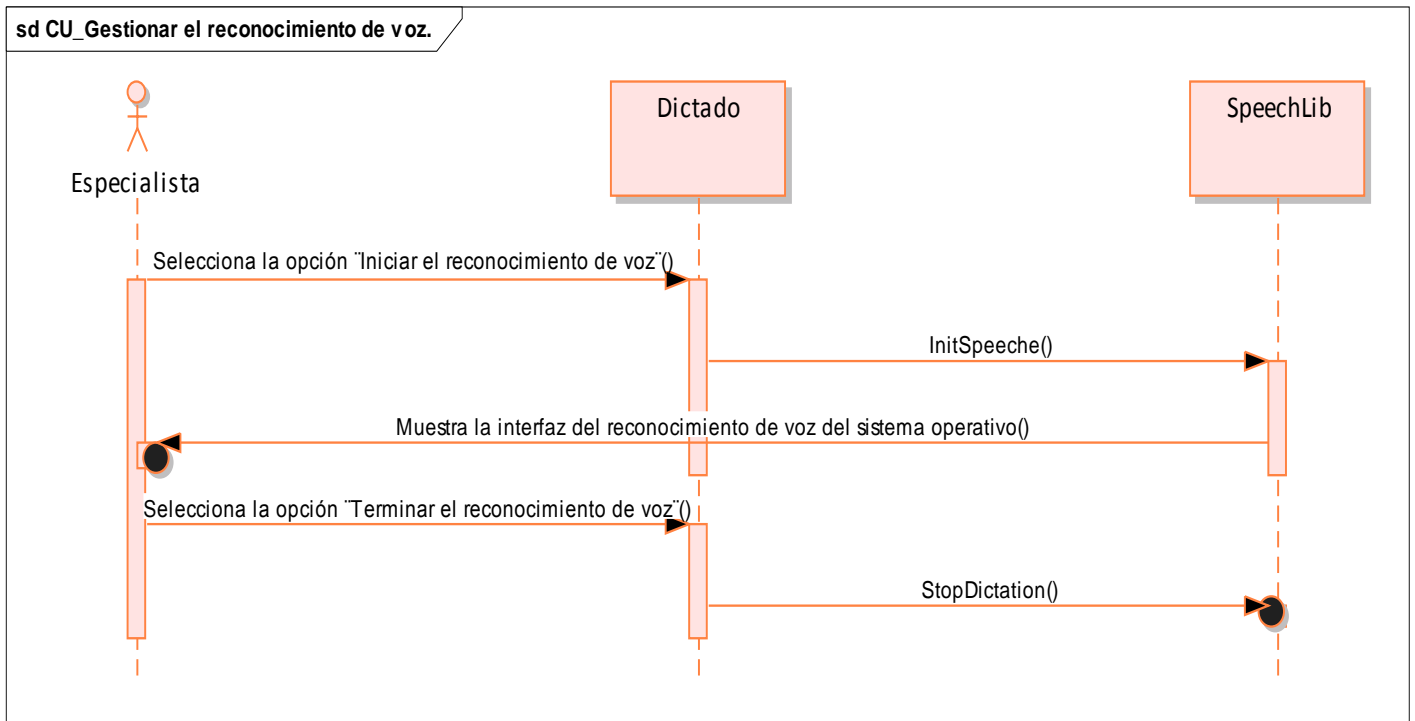
- + RNFO 1. Sistema Operativo Windows 7 o superior.
- + RNFO 2. Memoria RAM 512 MB o superior.
- + RNFO 3. Procesador Pentium IV 3.0 GHz o superior.
- + RNFO 4. Calidad de micrófono.

## Anexo 3 Diagrama de clases de los casos de uso arquitectónicamente significativos.



## Anexo 4 Diagrama de secuencia.

Diagrama de secuencia CU Gestionar el reconocimiento de voz.



Anexo 5 Descripción de las clases.

**Clase Dictation**

**Propósito:** Configura el reconocimiento de voz.

**Descripción**

<b>Nombre:</b> Dictation	
<b>Tipo de clase:</b> controladora	
<b>Atributo</b>	<b>Tipo</b>
_objRecoContext	SpRecoContext
<b>Para cada responsabilidad:</b>	
Nombre:	Dictation()
Descripción:	Inicializa los atributos.

# Componente para realizar el dictado automático en la elaboración de los informes diagnóstico. **Anexos**

Nombre:	InitSpeech()
Descripción:	Inicia el reconocimiento de voz.
Nombre:	StopDictation()
Descripción:	Termina el reconocimiento de voz.

Anexo 6 Tabla con las palabras comprobadas.

Palabras	Mostraron Dificultad
abacteriemia	
abdomen	
abdominalgia	
abducción	
abductor	X
ablación	
ablastémico	X
abraquia	X
absceso	X
acalasia	
alcacicocis	
acantoma	X
acardia	
acarofobia	
acedia	
acéfalo	
acefalocardia	X
acefalogastría	X
acefalopodia	X
acefaloquiria	X

acefalorraquia	
acefalotoracia	X
acelular	X
acestoma	X
acetilsalicílico	
acetosal	X
acidosis	
aclorhidia	X
acmé	X
acorea	X
acrílico	
actinocardiograma	X
actinoscopia	X
acuimetría	
acumetría	
adaquia	
adenectomía	
adenitis	
adenoide	
adenoidectomía	
adenoma	
adenopatía	
adenosis	
adh	
adicción	
adipoma	X
adiposis	X
adrenalina	
aducción	X

aductor	
aerobio	
aerosolterapia	X
afagia	X
afaquia	X
afasia	X
afección	
agenesia	
agluceemia	X
agorafobia	
agujetas	
alergia	
algia	X
alopecia	
alzheimer	
amigdalectomía	X
amigdalogloso	
amniocentesis	
amnioscopia	X
anabolismo	X
ancóneo	X
anemia	
anestesia	
aneurisma	
angioplastia	
anorexia	
angiografía	
anovulatorio	
apendicitis	



artroscopia	
audiometría	
auricular	
auriculares	
autorización	
axis	
aerobio	
bálsamo	
benigno	
biceps	
biliar	
bilirrubina	
bilis	
biología	
biopsia	
bioquímica	
biotipo	
boca	
bronquitis	
brazo	
bronquio	
bulimia	
cadáver	
calcio	
cálculo	
calentura	
calambre	
calcificación	
calor	

calmante	
caloría	
cáncer	
canino	
cánula	
capilar	
capula	
carcinógeno	
cardiaco	
carencia	
caries	
carótida	
cariotipo	
catabolito	
ceceo	X
catarata	
catarro	
catarata	
ceguera	
célula	
cerebelo	
cerebro	
cesárea	
cicatrices	
circulación	
circuncisión	
cirugía	
cisterna	