

Universidad de las Ciencias Informáticas

Facultad 6



Título: Video Sensor para el reconocimiento de palabras claves en audio.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores: Ana Ailín Borrego Cala.

Alexander Montero Cuadrado.

Tutora: Ing. Yunet Gasca Suárez.

La Habana, junio 2014

“Año 56 de la Revolución”

Pensamiento

*En lugar de ser un hombre de éxito, busca ser un hombre valioso: lo demás
llegará naturalmente.*

Albert Einstein

Declaración de autoría

Declaración de Autoría

Declaramos ser autores de la presente tesis que tiene por título: Video sensor para el reconocimiento de palabras claves en audio y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Ana Ailín Borrego Cala.

Firma del Autor.

Alexander Montero Cuadrado.

Firma del Autor.

Ing. Yunet Gasca Suárez.

Firma del Tutor.

Datos de Contacto

Datos de Contacto

Tutor:

➤ *Ing. Yunet Gasca Suárez*

Correo electrónico: ygasca@uci.cu

Graduada de Ingeniera en Ciencias Informáticas en el año 2013. Se desempeña como administradora de calidad en el Departamento de Señales Digitales del centro de Geoinformática y Señales Digitales (GEYSED).

Agradecimientos

Agradecimientos

Ana Ailín:

He de agradecer en el momento más importante de mi vida, en primer lugar a Dios por permitirme haber compartido estos años al lado de personas tan especiales. Con el permiso de todos los presentes y en especial de mi madre, agradezco por encima de todas las cosas a mi papá porque fue el motor impulsor de mi vida, quien desde muy niña me enseñó las cosas más importantes de la vida y quien aún hoy desde el cielo siento que día a día me apoya y me da fuerzas para seguir adelante, a él van dedicado todos mis logros. Agradezco a mi madre porque es por ella que hoy estoy aquí y porque lo ha sacrificado todo por verme triunfar. Agradezco a Raúl porque a pesar de todas las vicisitudes que he pasado nunca ha dejado de confiar en mí.

Quiero agradecer a mi tía que de tantos apuros me ha sacado en momentos tan difíciles, a mis hermanos Ana Delis, Jorgito y a ti Lay que tanta fuerza me has transmitido. Agradezco a Sonia mi otra madre incondicional, la persona que me acogió en el seno de su familia como si yo fuera una más de sus hijos sin poner condición alguna.

Un agradecimiento especial va dedicado a mis amigas de la universidad, Deylis por haberme dejado dormir contigo cuando por la noche tenía mucho miedo y por ayudarme con todos mis regueros, a Elizabet (la flaka) por haberme hablado bien fuerte cuando más lo necesité, a Tahimí por cargar conmigo cuando muchas veces me sentí sola y a Yohandi por escuchar siempre callado los chismes de nosotras las mujeres.

A ti mi amor, Osmín, un agradecimiento super especial porque en este último año has sido mi lucerito, me has apoyado en todo y me has ayudado sin condición. Cada lágrima que he derramado has sido tú quien la has secado y eso no lo voy a olvidar nunca. A Barbarita por tratarme como a una hija y por confiar en mí.

A mi tutora Yunet, quien pienso que en los próximos meses aún continúe soñando conmigo por lo mucho que la fastidié. A los profes Jean Michel y Solangel por lo mucho que me ayudaron en el transcurso de este trabajo.

A mi compañero de tesis porque a pesar de tantas incoherencias finalmente ha sabido tolerar mis resabios.

A todos los que de una forma u otra han tenido que ver con este logro sepan que hoy les agradezco de todo corazón y que nunca los olvidaré.

Agradecimientos

Alexander:

Y finalmente tras todo este tiempo, que la verdad, no se me ha hecho nada largo, me hallo escribiendo lo que es la culminación de la carrera universitaria. O tal vez no, nunca se sabe qué nos depara el destino. De todas formas es el momento de agradecer a todos aquellos que me han aportado algo en este interesante paseo que ha sido la Universidad.

Son muchas las personas que han formado parte de mi vida profesional a las que me encantaría agradecerles su amistad, consejos, apoyo, ánimo y compañía en los momentos más difíciles de mi vida.

De una manera especial a Ana Ailin mi compañera de tesis, por su esfuerzo durante toda la carrera y la entrega que le puso al desarrollo de nuestro trabajo de grado.

A los buenos profesores Solangel y Jean Michael que no podré dejar de agradecer su trabajo, con quien conté en todo momento para aclarar dudas y para orientarme, así como para transferir su conocimiento y hacer posible la realización de esta tesis.

Gracias a Miguel y Yoilan por compartir sus vidas a mi lado, sin sus ayudas no hubiera podido lograr llegar al final de esta aventura, ustedes son lo máximo.

A mi amigo Ramón por sus palabras, su confianza y su ayuda en todo momento, gracias por ser parte de esta travesía y enseñarme el verdadero significado de la amistad.

Gracias a Javier, Carlitos, Reynaldo, Pedro, Laffita por cada una de las palabras de aliento que me llenaron de fortaleza estoy muy orgulloso de ustedes.

A todos mis amigos de la universidad, del aula y a todos los que no están presentes entre nosotros hoy, que compartimos gran parte de la carrera y fueron de gran apoyo por tener siempre ganas de ayudar a todos.

De la misma manera como no agradecer al tribunal y al oponente que aportaron un granito de arena, por sus consejos, conocimientos y experiencias que me supieron transmitir en el momento preciso he indicado.

Y no puedo dejar de dar agradecimientos a Internet. Y qué es Internet más que las personas que dedican su tiempo en la red. En especial a las personas que dedican su tiempo libre a ayudar y a compartir lo que duramente han aprendido con los demás, como suele decirse, por amor al arte, porque ayudan a que el conocimiento sea de todos.

Dedicatoria

Dedicatoria

Ana Ailín:

Dedico este logro a mi padre porque aún desde el cielo continúa siempre estando a mi lado.

A mi madre porque sin su sacrificio nada hubiera sido posible.

Alexander:

A mi querida madre le dedico todo este triunfo, por haberme apoyado en todo momento, por sus consejos, valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por tu amor. Porque el orgullo que sientes por mí, fue lo que me hizo ir hasta el final. Por estar siempre a mi lado pendiente de mí en todo momento y fomentado en mí el deseo de superación y el anhelo de triunfo en la vida.

Gracias madre por tus palabras y ayuda, tú has sido pieza fundamental en mi vida y sin ti este sueño no se hubiera cumplido. Gracias por consentirme tanto y por enseñarme que con amor y paciencia se logran grandes cosas, gracias por enseñarme a no desmayar y luchar por mis sueños.

A mi linda hermanita por su cariño por ser un ejemplo a seguir para ella, y que escoja este camino que es el más correcto.

Dedico esta tesis en general a Mora, Mariela, Jesús por ser ejemplo de padres para mí por estar siempre dispuestos ayudarme y consentirme, gracias por todos sus detalles y por facilitarme tantas cosas durante mi carrera.

A mi tío Jorge que a pesar de la distancia ha ayudado a cumplir este gran sueño.

A ellos que siempre creyeron en mí y porque me sacaron adelante, dándome ejemplos dignos de superación y entrega, y porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles.

Resumen

Con el avance de las Tecnologías de la Información y las Comunicaciones (TICs) y el incremento de las exigencias por parte de los clientes con respecto a la seguridad de sus instalaciones, se ha logrado que los sistemas de video vigilancia alcancen un mayor auge. Debido a esto, el sistema de video vigilancia Xilema Suria tiene como prioridad potenciar su nivel de control y manipulación, garantizando índices menores de hechos delictivos.

La presente investigación tuvo como propósito desarrollar un video sensor para el reconocimiento de palabras claves en audio que permitiera el aprovechamiento del audio captado por el micrófono de las cámaras de video vigilancia. Este video sensor aumenta la confiabilidad del sistema Suria y reduce la necesidad de monitorear constantemente las cámaras por parte del personal involucrado.

El documento refleja todo el proceso de desarrollo del software, para el que fue necesario realizar un estudio de algunos sistemas desarrollados para el reconocimiento de voz. Se recogen las principales características de las herramientas utilizadas para la modelación e implementación de la solución. Se tratan los elementos fundamentales asociados a la metodología empleada y los principales artefactos generados por los flujos de trabajo que propone, además de la correcta implementación de la solución.

Palabras claves: video sensor, video vigilancia.

Abstract

With the advancement in the Information Technology and Communications and increasing demands from customers about the security of its facilities, the video surveillance systems had reach a greater importance. Because of this, the video surveillance system Xilema Suria has as a priority, to enhance the level of control and manipulation, ensuring lower rates of crimes.

The main objective of the present research was to develop a video sensor for recognizing audio keywords to allow the use of captured audio by the microphone of the video surveillance cameras. This video sensor increases Suria system reliability and reduces the need to the staff involved of constantly monitor the cameras.

The document reflects the entire software development process, it was necessary to conduct a study of some developed systems for speech recognition. The main features of the tools used for modeling and development of the solution was collected. Fundamental elements associated with the methodology and the main artifacts generated by workflows, in addition to the successful implementation of the solution are discussed.

Keywords: video sensor video surveillance.

Índice

Introducción.....	1
Capítulo 1: Fundamentación Teórica asociada al Reconocimiento del habla.	5
1.1. Introducción.....	5
1.2. Definiciones Generales	5
1.2.1. Video Sensor.....	5
1.2.2. Sistema de Video Vigilancia	5
1.2.3. Fonema	6
1.2.4. Sistema Basado en Conocimiento	6
1.3. Objeto de Estudio.....	6
1.3.1. Descripción General	6
1.4. Situación actual del dominio del problema.....	8
1.5. Soluciones existentes	9
1.5.1. Dragon Naturally Speaking	9
1.5.2. Windows Speech Recognition.....	10
1.5.3. Dictation Pro.....	11
1.5.4. Julius	11
1.5.5. Voce	12
1.5.6. Sphinx	12
Aporte de las soluciones estudiadas.....	12
1.6. Herramientas y Tecnologías	13
1.6.1. Metodología de Desarrollo.....	13
1.6.2. Lenguaje de Modelado	14
1.6.3. Herramienta de Modelado	15
1.6.4. Biblioteca Utilizada.....	15
1.6.5. Técnica de reconocimiento de voz empleada.....	16
1.6.6. Lenguaje de Programación.....	17

Índice

1.6.7.	Entorno de Desarrollo Integrado	18
1.6.8.	Framework de Desarrollo	19
1.1.	Conclusiones Parciales.....	20
Capítulo 2:	Análisis y Diseño.....	21
2.1.	Introducción.....	21
2.2.	Modelo de Dominio	21
2.2.1.	Descripción del Modelo de Domino.....	21
2.2.2.	Conceptos Asociados al Modelo de Dominio.....	22
2.3.	Requisitos Funcionales (RF).....	23
2.4.	Requisitos no Funcionales (RNF).....	23
2.4.1.	RNF1 Software.....	24
2.4.2.	RNF2 Hardware	24
2.4.3.	RNF3 Rendimiento	24
2.5.	Definición de Casos de Uso.....	24
2.5.1.	Actor del Sistema.....	24
2.5.2.	Diagrama de Caso de Uso del Sistema	25
2.5.3.	Descripción del Caso de Uso del Sistema	25
2.6.	Arquitectura de Software.....	27
2.6.1.	Estilo Arquitectónico	27
2.6.2.	Patrón Arquitectónico.....	28
2.6.3.	Patrones de Diseño	28
2.7.	Diseño de Software.....	32
2.7.1.	Diagrama de Clases del Diseño.....	32
2.8.	Diagramas de Interacción	34
2.9.	Conclusiones Parciales.....	35
Capítulo 3:	Implementación y Prueba	36
3.9.	Introducción.....	36

Índice

3.2. Modelo de Implementación.....	36
3.2.1. Implementación de Software	36
3.3. Diagrama de Despliegue	36
3.4. Diagrama de Componentes	37
3.5. Estándares de Codificación	38
3.5.1. Estilo de Codificación Utilizado	38
3.6. Pruebas	39
3.6.1. Pruebas de Caja Blanca	39
3.6.2. Pruebas de Rendimiento	45
3.6.3. Pruebas de Integración.....	49
3.6.4. Pruebas de Aceptación.....	49
3.6.5. Resultados de la pruebas de software	50
3.7. Conclusiones Parciales.....	50
Conclusiones Generales.....	51
Recomendaciones	52
Referencias	53
Bibliografía	55
Anexos	57

Índice de Tablas

Tabla. 1 Requisitos Funcionales	23
Tabla. 2 Actor del Sistema	24
Tabla. 3 Descripción del Caso de Uso: Transcribir de Audio a Texto	25
Tabla. 4 Descripción de las Clases del Diseño	33
Tabla. 5 Caminos Básicos de la funcionalidad Capturar	42
Tabla. 6 Caso de Prueba No1 de la funcionalidad Capturar	42
Tabla. 7 Caso de Prueba No2 de la funcionalidad Capturar	43
Tabla. 8 Camino Básico de la funcionalidad Reconocer	44
Tabla. 9 Caso de Prueba No1 de la funcionalidad Reconocer	44
Tabla. 10 Caso de Prueba No2 para la funcionalidad Reconocer	45
Tabla. 11 Palabras pronunciadas en el flujo de audio 1	46
Tabla. 12 Palabras pronunciadas en el flujo de audio 2	47
Tabla. 13 Palabras pronunciadas en el flujo de audio 3	47
Tabla. 14 Análisis del flujo de audio 4	48
Tabla. 15 Algoritmos fundamentales implementados	58

Índice de Figuras

<i>Fig. 1 Modelo de Dominio</i>	22
<i>Fig. 2 Diagrama de Caso de Uso del Sistema</i>	25
<i>Fig. 3 Evidencia de patrón Experto</i>	28
<i>Fig. 4 Evidencia del patrón Creador</i>	29
<i>Fig. 5 Evidencia del patrón Controlador</i>	30
<i>Fig. 6 Evidencia del patrón Alta Cohesión</i>	30
<i>Fig. 7 Evidencia del patrón de Bajo Acoplamiento</i>	31
<i>Fig. 8 Evidencia del patrón interfaz</i>	32
<i>Fig. 9 Diagrama de Clases del Diseño</i>	33
<i>Fig. 10 Diagrama de Secuencia Genérico</i>	34
<i>Fig. 11 Diagrama de Despliegue</i>	37
<i>Fig. 12 Diagrama de Componentes</i>	38
<i>Fig. 13 Funcionalidad Capturar</i>	41
<i>Fig. 14 Grafo de flujo de la funcionalidad Capturar</i>	41
<i>Fig. 15 Funcionalidad Reconocer</i>	43
<i>Fig. 16 Grafo de flujo de la funcionalidad Reconocer</i>	44

Introducción

En diversos ámbitos empresariales son incorporados cada vez con más frecuencia, sistemas de video vigilancia, con el objetivo de satisfacer sus necesidades de seguridad o de control de la actividad laboral. El empleo de estos sistemas ha sido concebido como mecanismo de respuesta temprana ante situaciones que pongan en peligro recursos humanos y materiales. Es por ello que en los últimos años el desarrollo tecnológico, la demanda de mayores niveles de seguridad y el estudio de técnicas de análisis de video traen consigo un gran paso de avance en este campo. Como resultado de los estudios que consecuentemente provoca dicho avance, se ha visualizado la presencia de los sistemas de video vigilancia empleando cámaras de red o cámaras IP¹. Dichas cámaras presentan funcionalidades que le brindan a los mismos mayor seguridad y protección de las instalaciones para los que son creados, ya que, es de vital interés que su confiabilidad aumente en la medida de lo posible, evitando así la mayor cantidad de delitos. (Mata, 2010)

Entre dichas funcionalidades se encuentra el procesamiento sobre el flujo de video, el cual brinda la ventaja de emitir respuestas programadas. Con ello facilita su uso e incluye funciones integradas como la detección de movimiento, alarma anti manipulación y alarma ante la detección de audio. Este último elemento, al ser analizado, puede garantizar que la apertura indebida de una puerta o determinadas palabras pronunciadas notifiquen una señal de alerta al sistema y con ello contribuyan a su seguridad (Fernández, 2013).

Al hacer uso del flujo de audio captado e identificar palabras claves es indispensable el procesamiento de la voz, el cual puede ser empleado para identificar a una persona o transcribir dicho audio a texto. La transcripción a texto tiene una importancia significativa debido a que esta tarea, por lo general, al ser realizada de forma manual, es asignada a personas que, aunque pueden poseer los conocimientos suficientes, experiencias, o herramientas necesarias para obtener un resultado con calidad, les es difícil lograrlo en un corto espacio de tiempo (Márquez, 2006).

¹ IP: *Internet Protocol*.

Introducción

Es por ello que analizando lo antes planteado se puede establecer un vínculo entre este proceso (transcripción de audio a texto) y un sistema de video vigilancia con el objetivo de añadir funcionalidades al mismo que favorezcan la reducción de incidentes de inseguridad.

En el sistema de Video Vigilancia Xilema Suria existen un conjunto de módulos que operan juntos y cada cual cumple con una tarea diferente. El módulo de Análisis se encarga de acoplar un conjunto de video sensores² facilitando la labor de vigilancia y aportando inteligencia extra al sistema. Este módulo mantiene comunicación con el gestor³, es decir, las operaciones que en él se realizan sobre el flujo de video permiten la detección de algún evento significativo y brindan la posibilidad de enviar un mensaje de alerta ante la ocurrencia de alguna anomalía. En este proceso de análisis sobre el video es donde se centra la presente investigación, ya que en dicho sistema se pueden observar todas las cámaras al mismo tiempo pero solo se puede escuchar una en específico. Escuchar todas simultáneamente provocaría un solapamiento de voces. Esta situación crea una limitante para la persona encargada de la manipulación y el funcionamiento del sistema, debido a que puede estar siendo indiferente a la ocurrencia de algún delito, que a simple vista se asemeja a una situación común.

Teniendo en cuenta los aspectos antes mencionados se define como **problema de la investigación**: ¿Cómo contribuir a video vigilancia utilizando como medio el audio, para su aprovechamiento en el sistema de Video Vigilancia Xilema Suria?

La investigación en curso tiene como **objeto de estudio**: las técnicas de reconocimiento del habla. Se determina como **campo de acción**: el reconocimiento de palabras claves en los videos obtenidos de cámaras IP. De ahí que, para dar solución al problema, el **objetivo general** de la investigación sea: Desarrollar un video sensor que permita el reconocimiento automatizado de palabras claves en audio para el sistema de Video Vigilancia Xilema Suria

Se plantean entonces como **preguntas científicas**:

² Video Sensor: Algoritmo para identificar variaciones en la secuencia de imágenes de un video haciendo algún tipo de procesamiento sobre el mismo (Bhanu, y otros, 2011).

³ Gestor del Sistema de Video Vigilancia Xilema Suria: Modulo encargado de orquestar toda la comunicación entre los diferentes módulos del sistema en cuestión.

Introducción

1. ¿Cuáles son los fundamentos teóricos del proceso de reconocimiento de palabras claves en audio mediante un video sensor?
2. ¿Cuáles son las características que debe cumplir el sistema para reconocer palabras claves en audio mediante un video sensor?
3. ¿Cómo estructurar el proceso de desarrollo del sistema para que permita el reconocimiento de palabras claves en audio?
4. ¿El sistema desarrollado permite reconocer palabras claves en audio mediante el flujo de video captado por cámaras IP del sistema Video Vigilancia Xilema Suria?

Las **tareas de la investigación** trazadas para resolver el problema y alcanzar el objetivo planteado son:

1. Caracterización de los procesos relacionados con el reconocimiento de palabras claves en audio.
2. Descripción del estado del arte a nivel nacional e internacional de las herramientas existentes para reconocer palabras claves en audio.
3. Caracterización de las herramientas y tecnologías a utilizar en el desarrollo del sistema de reconocimiento de palabras claves en audio mediante un video sensor.
4. Análisis de la situación existente para generar los artefactos que se involucran en la construcción del software.
5. Diseño de los artefactos que se generan en la construcción del software.
6. Implementación de la herramienta que posibilite mediante un video sensor, el reconocimiento de palabras claves en audio.
7. Validación de la propuesta de solución para el reconocimiento de palabras claves mediante un video sensor.

Para establecer una orientación y dirección adecuada en la investigación y con el propósito de dar solución a las tareas propuestas se utilizan varios **Métodos Científicos**, entre ellos se encuentran:

Introducción

Métodos Teóricos, se utilizó en la presente investigación el Método Analítico – Sintético, analizando la situación existente en el Sistema de Video Vigilancia Xilema Suria con respecto al trabajo con el audio captado por las cámaras del sistema y para realizar una síntesis del problema con el objetivo de alcanzar una solución viable. Se empleó el Método Histórico-Lógico para estudiar la trayectoria del reconocimiento de voz desde sus inicios con el objetivo de reproducir en el plano teórico lo más importante de este fenómeno y estudiar sus posibles aplicaciones para la integración al sistema de Video Vigilancia Xilema Suria.

Como técnica de recopilación de información en esta investigación se utilizó La Entrevista de tipo no estructurada, mediante preguntas directas al líder del proyecto Video Vigilancia Xilema Suria. Con el propósito de obtener un conocimiento detallado de la situación del proyecto con respecto al posible reconocimiento de voz a través de un video sensor. **(Ver Anexo 1)**

La investigación en curso está distribuida en 3 capítulos. En el primer capítulo titulado: Fundamentación Teórica asociada al Reconocimiento del habla, se exponen las definiciones asociadas al reconocimiento automatizado de voz y se analizan las tendencias y tecnologías existentes con respecto al dominio del problema. Se definen las herramientas y lenguaje de modelado a utilizar. Se realiza una breve descripción de la metodología a emplear, con el objetivo de modelar y desarrollar la propuesta de solución. Además se plantea una síntesis detallada del objeto de estudio con la intención de explicar cómo ha transcurrido el proceso de reconocimiento automatizado de palabras claves en audio.

En el segundo capítulo se realiza el análisis y diseño del sistema a desarrollar. Se plantean los requisitos funcionales y no funcionales que guiarán la investigación, así como el modelo de dominio describiendo los procesos fundamentales existentes en el sistema de Video Vigilancia Xilema Suria. Se establece la arquitectura y se selecciona el patrón arquitectónico en función de sentar las bases para la construcción del sistema. Se describen los casos de uso que agrupan las principales funcionalidades con las que debe cumplir la solución propuesta y son confeccionadas las clases de análisis y los diagramas de interacción.

En el capítulo 3 se aborda todo el proceso de construcción y pruebas para validar la propuesta de solución. Se describe el modelo de implementación que incluye los diagramas de despliegue y componentes del sistema. Se le realizan pruebas de caja blanca y de rendimiento al sistema en función de comprobar el cumplimiento de los requisitos funcionales que fueron capturados durante el análisis.

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

Capítulo 1: Fundamentación Teórica asociada al Reconocimiento del habla.

1.1. Introducción

En el presente capítulo se hace referencia a las principales definiciones que servirán como punto de partida para un mayor entendimiento por parte de usuarios y desarrolladores del sistema a construir. Se realiza un estudio de sistemas que poseen soluciones similares, teniendo en cuenta el estado del arte de dichos sistemas tanto a nivel nacional como internacional. Se describen las tendencias y tecnologías actuales así como la metodología utilizada para el modelado de los procesos que serán definidos. Se explica en detalle el objeto de estudio propuesto y se dejan en claro las herramientas y tecnologías seleccionadas para el desarrollo de la aplicación, aprovechando las ventajas que estas poseen para ofrecer una rápida solución al problema planteado.

1.2. Definiciones Generales

1.2.1. Video Sensor

Algoritmo que permite procesamiento de algún tipo sobre un flujo de video para identificar variaciones significativas en la secuencia de imágenes (Bhanu, y otros, 2011).

1.2.2. Sistema de Video Vigilancia

Un sistema de video vigilancia, según la empresa *Accesor Applications and Services*, es un conjunto de componentes (cámaras de video vigilancia) enlazados donde a diferencia de la televisión convencional este es un sistema pensado para un número reducido de espectadores. Puede estar compuesto por una o más cámaras de video vigilancia conectadas a uno o más monitores o televisores (*Accesor Applications and Services*, S.A., 2014).

De lo antes expuesto se puede concluir que un sistema de video vigilancia consiste en una instalación de cámaras de video, donde las imágenes se registran en un grabador y pueden ser vistas en un monitor con el fin de garantizar la seguridad y protección de diversas instalaciones. Los sistemas de video vigilancia son sencillos de utilizar ya que se gestionan de forma similar a un video doméstico.

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

1.2.3. Fonema

Un fonema es la unidad menor de sonido en un idioma. Es un sonido con sentido que cambia una palabra en otra (García, 1999).

Alberto Bustos plantea que la definición de fonema está referida a unidades de análisis lingüísticos que están basados en los sonidos de una lengua pero que no se debe confundir con estos. Un fonema es el segmento mínimo al que se llega por abstracción a partir de los sonidos de la cadena hablada que es capaz de sustentar una distinción de significado careciendo de significado el mismo (Bustos, 2011).

A partir de lo anterior se puede concluir que, el fonema es cada sonido que distingue significados en una lengua. Es la unidad fónica menor e indivisible del lenguaje.

1.2.4. Sistema Basado en Conocimiento

Los sistemas expertos o sistemas basados en conocimiento son sistemas que tienen como finalidad la correcta reproducción del comportamiento humano en su dominio de competencia. Estos sistemas son elaborados con técnicas de Inteligencia Artificial que, de igual forma que el experto humano al que intenta emular, resuelve los problemas complejos y difíciles que circunscriben a un dominio específico y delimitado (García, 2012).

Otro concepto relacionado con el tema plantea que: los sistemas expertos son una estrategia orientada a las personas que permite la explicación del conocimiento de los expertos en la ejecución de una determinada tarea (Paniagua, y otros, 2007).

Con lo antes expuesto se puede determinar que, un sistema experto o sistema basado en conocimiento se define como: un sistema que simula el conocimiento humano para resolver los problemas de forma similar a lo que estos harían.

1.3. Objeto de Estudio

1.3.1. Descripción General

El desarrollo de sistemas de video vigilancia es consecuente con la intención de aumentar los índices de seguridad ciudadana. Con ellos se busca reducir en la medida de lo posible las pérdidas materiales y los

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

incidentes de inseguridad. El hecho de poder integrar a estos un sistema para el reconocimiento de voz, ofrece la ventaja de minimizar la vigilancia humana de los monitores y además, el audio captado por el micrófono de las cámaras es aprovechado con el fin de no malgastar dicho recurso. Esto permite al sistema hacer un análisis exhaustivo del flujo de texto transcrito con la posibilidad de emitir un mensaje de alerta ante la detección de alguna anomalía.

Para trabajar sobre esta línea se debe tener en cuenta que el reconocimiento de voz no es más que un sistema que tiene almacenados fonemas y estructuras de sílabas que pueden ser pronunciadas. Estas señales de voz pronunciadas o emitidas por un individuo son procesadas por un sistema de reconocimiento automático del habla lo que permite realizar varios procesos y denotar su comportamiento, además de transcribir la misma a texto. De ahí que el proceso de reconocimiento de voz se vea inmerso en dos bloques: entrenamiento y reconocimiento. El entrenamiento es el pilar más complicado del proceso, realizándose generalmente partiendo de un Sistema Basado en Conocimiento (SBC). Para ello pueden utilizarse varias técnicas de reconocimiento del habla, como: Codificación Predictiva Lineal, Redes Neuronales Artificiales y Modelos Ocultos de Markov. (Rodríguez, 2006)

Codificación Predictiva Lineal (LPC)

- Es una de las técnicas más utilizadas en el procesamiento de señales de audio y procesamiento de voz según (Grijalva, 2003). Plantea además que LPC supone que la señal de voz la produce una especie de zumbador en el extremo de un tubo, con sonidos adicionales. Analiza dicha señal realizando una estimación de los elementos que la conforman: la eliminación de sus efectos partiendo de la señal de la voz, la estimación de la intensidad y la frecuencia del zumbido restante. El proceso de eliminación de residuos se denomina filtrado inverso y la señal que queda después de la sustracción de la señal modelada filtrada se llama residuo.

Redes Neuronales Artificiales

- Se construyen para simular el funcionamiento del cerebro humano. Están constituidas por una red de neuronas electrónicas que cambian su configuración a medida que el sistema aprende. Se debe tener en cuenta, además, que al utilizar modelos de redes neuronales, se aprecia que están compuestos por varios nodos en su forma simple, que operan en paralelo y que a su vez están arreglados en patrones con la intención de simular redes neuronales biológicas. Dichos nodos almacenan los pesos asociados a las

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

conexiones y suman un número n de entradas que se multiplican por el peso de la conexión correspondiente. Al resultado obtenido se le multiplica una función de activación no lineal. Las redes neuronales para el reconocimiento de voz convierten dicha señal a representación fonética y luego a texto. Cabe destacar que haciendo uso de ellas no se obtienen resultados lo suficientemente convincentes (Bravo, y otros, 2000).

Modelos Ocultos de Markov

- Describen un proceso de probabilidad el cual, produce una secuencia de eventos o símbolos observables. Se les llama modelos ocultos debido a que paralelo a ellos existe un proceso de probabilidad que no es observable pero que si influye en la secuencia de eventos observados. Es el mejor algoritmo para reconocer una palabra al tiempo que es procesada. Este está constituido por una especie de cadena que se concatena para formar palabras o frases continuas. Con ellos se le pretende dar solución a tres problemas fundamentales: En primer lugar, se desea que, dada una secuencia de observación y un modelo, se pueda hacer un cálculo eficiente de la probabilidad de secuencia de observación producida por el modelo. Además es bien importante tener en cuenta, como segundo aspecto, que dada una secuencia de observación y un modelo cualquiera se establezca una secuencia de estados correspondientes que sea eficiente en algún sentido. El tercer problema a resolver se centra en cómo ajustar los parámetros del modelo para describir de una mejor forma cómo se construye una secuencia de observación (Díaz Verdejo, y otros, 2012).

El reconocimiento de voz automatizado se basa en el reconocimiento de patrones y de sistemas biométricos. La voz tiene que ser indispensablemente analizada para la extracción de información que puede ser de gran importancia, una vez que se ha digitalizado. Vale la pena destacar que aspectos como: tamaño del vocabulario, sistemas dependientes e independientes del locutor, voz aislada, discontinua y continua, voz aplicada a tareas, voz leída o espontánea y las condiciones adversas que pueden ocurrir son aspectos importantes a tener en cuenta en el proceso de reconocimiento del habla. (Rodríguez Oropeza, 2006)

1.4. Situación actual del dominio del problema

A continuación se presenta la situación actual por la que atraviesa el sistema de Video Vigilancia Xilema Suria.

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

El sistema en cuestión tiene su enfoque fundamental en la seguridad y protección de las instalaciones que lo utilizan. Su módulo de Análisis es pieza clave en este proceso pues está compuesto por video sensores que garantizan el procesado del flujo de video, en función de detectar la ocurrencia de alguna anomalía y notificar al gestor. Dichos video sensores están relacionados con: la detección de tapado de las cámaras, reconocimiento facial, detección de pérdida de foco de las cámaras, entre otros. Todos estos procesos contribuyen a la creación de un sistema robusto, analizando las variaciones (movimientos humanos y de objetos) en la secuencia de movimientos que ocurren en un entorno variado. En determinado momento se hace imposible escuchar el audio que transmiten todas las cámaras al mismo tiempo. Esto trae consigo que puedan ser ignorados delitos claves que afecten el propósito con el que fue creado el sistema. Además ante esta situación el audio captado por las cámaras es completamente desechado malgastando recursos en los que ya se ha invertido y aumentando el desgaste físico de la persona encargada del correcto funcionamiento del sistema.

La implantación de un video sensor para el reconocimiento de palabras claves en audio garantizaría que Xilema Suria sea capaz de un mayor control minimizando la vigilancia humana de los monitores. Por otra parte permitiría que los hechos delictivos fueran detectados con más facilidad, aun cuando los protagonistas de estos puedan eludir la visibilidad de las cámaras.

1.5. Soluciones existentes

Como resultado del estudio de distintas bibliografías se han encontrado varios software que realizan el reconocimiento de frases usando el audio como medio, como primera muestra de ello se puede mencionar a:

1.5.1. *Dragon Naturally Speaking*

Es un sistema de reconocimiento de voz que permite dar órdenes a la computadora. Se habla y *Dragon Naturally Speaking* transcribe lo que se dice, aspecto que es similar a lo que se pretende en la presente investigación. Este software utiliza una interfaz de usuario mínima y las palabras aparecen en un texto de ayuda flotante. El programa tiene tres áreas principales de funcionalidad: el dictado, convertir de texto a voz y la entrada de comandos. Además, a los perfiles de voz se puede acceder a través de diferentes equipos en un entorno de red, aunque el hardware de audio y la configuración deben ser idénticos en

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

ambas computadoras. La versión *Professional* permite la creación de comandos personalizados para controlar los programas o funciones no integradas en *Dragon Naturally Speaking*.

Este es un software privativo lo que dificulta su utilización para software libre y en su implementación se utilizan Modelos Ocultos de Markov (Iborra, 2008).

1.5.2. *Windows Speech Recognition*

Software privativo creado específicamente para *Windows Vista*, siendo capaz del reconocimiento automático de voz para que los usuarios puedan interactuar con sus ordenadores usando solamente la voz. Es diseñado pensando en usuarios deseosos del ahorro significativo del teclado y el mouse de la computadora. Se le pueden dictar documentos y correos electrónicos en aplicaciones de uso común, utilizar comandos de voz para iniciar y cambiar entre aplicaciones, controlar el sistema operativo, e incluso llenar los formularios de la Web.

Este software es construido utilizando las últimas tecnologías de voz de *Microsoft Windows Vista*. Proporciona una buena precisión en el reconocimiento que mejora con cada uso, ya que se adapta a su estilo y el vocabulario de habla. Su disponibilidad se encuentra en idiomas como: inglés (EE.UU.), inglés (Reino Unido), alemán (Alemania), francés (Francia), español (España), japonés, chino (tradicional) y chino (simplificado).

Algunas de las características más significativas que posee son:

- 1- Corrección: Solucionar de manera eficiente las palabras incorrectamente reconocidas por la selección de alternativas de formulario para la frase o palabra dictada o de ortografía de la palabra.
- 2- Tutorial interactivo: El tutorial de reconocimiento de voz interactivo que enseña cómo utilizar *Windows Speech Recognition*.
- 3- Desambiguación: Resolver fácilmente las situaciones ambiguas con una interfaz de usuario para una aclaración. Cuando se dice un comando que se puede interpretar de muchas maneras, el sistema aclara lo que pretende (Iborra, 2008).

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

1.5.3. Dictation Pro

Dictation Pro es un software de dictado y transcripción profesional diseñado para *Windows 7*. El programa al ser instalado requiere un ejercicio de entrenamiento de 10 minutos. Esto al mismo tiempo le enseña cómo utilizar el programa mientras aprenden a reconocer y registrar sus patrones únicos de voz y expresión. De esta forma, puede transcribir con precisión sus palabras. Cuenta con múltiples capacidades para garantizar que cada persona que utiliza el programa tendrá una experiencia de calidad. Cuando se utiliza *Dictation Pro*, se pueden editar los documentos mediante comandos de voz o introducirlos manualmente. Se edita en la plataforma de procesamiento de texto incluido. Incluso tiene la capacidad de crear comandos de voz personalizados para garantizar una experiencia de dictado más suave. Este software emplea los Modelos Ocultos de Markov para el reconocimiento de voz a través del micrófono de la computadora. (*Microsoft Windows*, 2011)

1.5.4. Julius

Julius es un motor de reconocimiento de voz continua con vocabulario de alto rendimiento (Iborra, 2008). El reconocimiento de las palabras está basado en N-gramas y en Modelos Ocultos de Markov dependientes del contexto. En ocasiones realiza decodificación sobre tareas de dictado en la mayoría de las computadoras actuales prácticamente en tiempo real.

La biblioteca usa técnicas de búsqueda eficiente para reconocer palabras. Varios de los Modelos Ocultos de Markov son soportados como “trifonos⁴” de estado compartido y también se permite vincular modelos con cualquier número de mezclas, estados o fonemas. La principal plataforma es Linux y estaciones de trabajo Unix, aunque también está disponible para Windows. La versión más reciente se desarrolla sobre Linux y *Windows* (cygwin / MinGW), y también tiene versión para *Mackintosh*. Julius se distribuye con licencia abierta junto con el código fuente. Ha sido desarrollado como un software de investigación para el idioma japonés desde 1997 aunque también puede ser configurado para el inglés. El trabajo fue continuado por IPA (*Japanese Dictation Toolkit*) Project (1997-2000), *Continuous Speech Recognition Consortium*, Japan (CSRC) (2000-2003) y actualmente por *Interactive Speech Technology Consortium* (ISTC) (Iborra, 2008).

⁴ Trifonos: Modelo que constituye una unidad básica de reconocimiento compuesto por tres fonemas.

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

1.5.5. Voce

Es una biblioteca multiplataforma para síntesis y reconocimiento de voz en italiano. Es accesible desde Java y C++ y utiliza Sphinx4 como biblioteca importada. El consumo de esta biblioteca hará que el usuario mantenga una interacción de discurso ambigua con su computadora. Dentro de las ventajas que presenta la misma se puede mencionar que su interacción está pensada en personas con discapacidades físicas relacionadas con la ceguera y el desorden del control corporal. *Voce* no requiere ningún entrenamiento especial y los usuarios pueden circular a una distancia determinada de la PC⁵ y aun así seguir interactuando con ella. La desventaja de esta aplicación con respecto a su uso en la presente investigación es que está implementada solo para el italiano y no puede ser configurada para el español (Center, 2011)

1.5.6. Sphinx

Aprovechando la ventaja de que esta biblioteca es un armazón para el reconocimiento de voz, puede usarse como generador o sintetizador de voz. Además tiene licencia de código abierto, por lo cual está disponible para toda la comunidad de desarrolladores. Existen dos distribuciones de *Sphinx* una con las bibliotecas necesarias para usar el reconocedor de voz en aplicaciones propias y otra que incluye además todo el código fuente para desarrolladores que deseen realizar modificaciones, siendo la primera la de mayor importancia para la investigación.

Sphinx ofrece tres bibliotecas de extensión .jar con tres modelos acústicos diferentes. Con esta herramienta se puede crear una biblioteca propia con un modelo acústico creado por la propia persona. Para el reconocimiento automático de voz esta biblioteca emplea los Modelos Ocultos de Markov por ser la técnica que reconoce la señal de la voz de manera más exacta (Iborra, 2008).

Aporte de las soluciones estudiadas

Al realizar un estudio del estado del arte tanto a nivel nacional como internacional de los principales software existentes para el reconocimiento de voz utilizando el audio como medio, se pudo constatar que: las aplicaciones antes mencionadas no aportan una solución factible en el desarrollo de la presente

⁵ PC: *Personal Computer*

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

investigación. Aunque es importante mencionar que las tecnologías empleadas por ellas si son de gran ayuda. Se pudo conocer gracias a estas que dentro de las técnicas que existen para el reconocimiento de voz la más empleada es la de los Modelos Ocultos de Markov y que es la que ha arrojado resultados más significativos. Este aspecto puede servir como guía en la presente investigación para lograr reconocer la voz presente en el audio captado de un flujo de video.

1.6. Herramientas y Tecnologías

Antes de comenzar el desarrollo de la solución que se propone, una de las primeras tareas a ejecutar es definir las herramientas y tecnologías existentes a utilizar en el desarrollo de la presente investigación. Puesto que las mismas permiten guiar, visualizar, entender y documentar el proceso de desarrollo de software.

1.6.1. Metodología de Desarrollo

Metodología de desarrollo de software es el ámbito de trabajo donde se estructura, planifica y se tiene el control del proceso de desarrollo en sistemas de información. Son una serie de pasos a seguir que ayudan a la documentación para desarrollar productos de software. Es vigente la utilización de dichas metodologías para dar soluciones a los desarrolladores por su importancia para la gestión de proyectos de forma eficiente. Pueden ser clasificadas como metodologías ágiles y pesadas (Rumbaugh, y otros, 2000).

Proceso Unificado de Desarrollo de Software(RUP)

Es un proceso de desarrollo de software que ha resultado ser bien difundido y usado para analizar, documentar e implementar proyectos de software. Su ciclo de vida implementa el Desarrollo en Espiral y organiza las tareas en fases e iteraciones. Posee una estructura dinámica que posibilita que sea un proceso altamente interactivo y he aquí la descripción de las fases antes mencionadas con las que cuenta para esto.

- 1. Comienzo o Inicio:** Se realiza una descripción del negocio y se le pone límites al proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- 2. Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen.

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

- 3. Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varios *release* (liberaciones) del producto que han pasado las pruebas. Se ponen estos *release* a consideración de un subconjunto de usuarios.
- 4. Transición:** El *release* ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores (Rumbaugh, y otros, 2000).

¿Por qué utilizar RUP?

Se decide emplear RUP como metodología de desarrollo debido a que el sistema de Video Vigilancia Xilema Suria la utiliza para la implementación de soluciones y la que se pretende desarrollar constituye una contribución a dicho proyecto, así consta en el expediente del mismo. Es importante destacar además que como se pretende definir áreas del conocimiento para la ingeniería de software y se proponen procesos pre-definidos como base para cada una de ellas, esta metodología será de gran ayuda.

1.6.2. Lenguaje de Modelado

Modelar un software es una técnica para aminorar la complejidad indiscutible que estos poseen. Un modelado eficiente ayuda al ingeniero de software a visualizar lo que pretende construir, además de que sirve para la comunicación con el cliente de una manera amena. En la investigación en curso se hará uso del lenguaje de modelado *Unified Modeling Language (UML)*.

UML (Unified Modeling Language) 2.0

Es el lenguaje de modelado que emplea la metodología RUP para modelar sus negocios. Estructura la aplicación, su comportamiento y su arquitectura. *Unified Modeling Language* o Lenguaje Unificado de Modelado (UML) es usado para describir sistemas orientados a objetos, con la facilidad de que es simple de aprender y flexible (Rumbaugh, y otros, 2000).

¿Por qué emplear UML 2.0?

Se decide para la especificación de los artefactos generados por la metodología escogida en la presente investigación, el empleo de UML como lenguaje de modelado porque es el que predefine la misma (RUP).

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

Se aprovecha con ello la ventaja de que UML brinda la posibilidad del modelado del software desde la fase de inicio hasta la implementación.

1.6.3. Herramienta de Modelado

La herramienta de modelado es lo que le proporciona al ingeniero la posibilidad de automatizar actividades manuales y mejorar su visión general de la ingeniería. Ayudan a garantizar que la calidad se diseñe antes de llegar a construir el producto (Pressman, 1997). Una de las herramientas que emplean el lenguaje de modelado a utilizar es *Visual Paradigm*, precisamente la usada en la presente investigación.

Visual Paradigm para UML 8.0

Visual Paradigm para UML es una herramienta para el desarrollo de software utilizando modelado con UML en, creación de sistemas de gran importancia y que necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos (S.A., 2013). Dentro de sus más relevantes características está que es de código abierto lo que facilita su utilización en el desarrollo de la aplicación que se pretende construir.

¿Por qué emplear Visual Paradigm 8.0?

Se decide utilizar *Visual Paradigm* en su versión 8.0 para el modelado de software, teniendo en cuenta que *Visual Paradigm* es una potente herramienta en cuanto al empleo de navegación entre la escritura de código y su visualización. Con dicha herramienta se tiene un ambiente superior en cuanto a modelado y permite la sincronización del código fuente en tiempo real. *Visual Paradigm* es multiplataforma, lo que facilita el uso de dicha herramienta tanto en software libre como privativo. Posee capacidades de ingeniería inversa y directa, es fácil de instalar y de utilizar, además de que con él se pueden generar diagramas de casos de uso, secuencia y diagramas de clases lo que facilita en gran medida la construcción de software para el reconocimiento de frases en audio.

1.6.4. Biblioteca Utilizada

Una biblioteca en el desarrollo de software es un conjunto de subprogramas que contienen códigos y datos que sirve para ofrecer servicios a otros programas y ayuda a los programadores a desarrollar el software. No necesita ser modificado y el código que contienen, se añade al programa principal cuando se genera (Iborra, 2008).

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

Sphinx4 v4.5

Luego de un amplio estudio de las soluciones existentes que pueden ser empleadas como ayuda en el desarrollo de la presente investigación, se determinó que será usada la biblioteca Sphinx4 para la implementación del reconocimiento de palabras claves en audio. Esta biblioteca es de código abierto y contiene las funcionalidades necesarias para el reconocimiento automatizado de voz. De ella se utilizan clases que son fundamentales en el video sensor para el reconocimiento de palabras claves en audio como:

- **Configuration:** Se encarga de cargar el modelo acústico, el modelo de lenguaje y el diccionario a emplear. Además determinando que componentes van a ser usados por el sistema y especificando detalladamente la configuración de cada uno de esos componentes.
- **StreamSpeechRecognizer:** Es el motor que inicia la interfaz de entrada y crea el resto de los componentes asociando a él el fichero de audio de entrada.
- **SpeechResult:** Contiene todos los caminos que han alcanzado el estado final.

¿Por qué Sphinx4 v4.5?

Se decide en el desarrollo de la investigación en curso el uso de la biblioteca Sphinx4 debido a que es de código abierto lo que permite hacer modificaciones en el mismo. De las estudiadas, esta es la biblioteca que permite ser configurada para que el reconocimiento automatizado de la voz se realice en idioma español. Con *Sphinx4*, además, se puede realizar el análisis de un flujo de audio captado por el micrófono de las cámaras que transmiten el flujo de video en un entorno de video vigilancia.

1.6.5. Técnica de reconocimiento de voz empleada

Una vez estudiadas las técnicas para el reconocimiento automatizado de voz se pudo constatar que algunas de ellas como LPC y Redes Neuronales no son idóneas para este proceso. En el caso específico de la Codificación Predictiva Lineal, no es empleada en la investigación en curso pues en el reconocimiento digital de señales ha probado ser más eficiente para procesar señales producidas por instrumentos musicales (Rodríguez, 2006). Las Redes Neuronales para el trabajo en curso tendrían un número considerable de entradas y los cálculos que ellas usan y que son relativamente fáciles, se harían

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

muy complejos. Debido a ello se emplea en la presente investigación la técnica de Modelos Ocultos de Markov pues en estudios realizados ha demostrado ser más exacta en el reconocimiento de voz (Rodríguez, 2006). Es escogida esta técnica además porque la biblioteca a utilizar para el desarrollo de la aplicación emplea la misma como modelo a seguir para el reconocimiento de voz. De esta forma se pretende dar solución al problema planteado sobre el reconocimiento del habla mediante la técnica que ha arrojado resultados más satisfactorios.

1.6.6. Lenguaje de Programación

C++ en su estándar ANSI 98

Es la extensión del lenguaje C que permite a los programadores el trabajo con objetos y la programación estructurada. Es un lenguaje variado, muy potente y su uso posibilita la programación a alto nivel. Con él se pueden programar desde sistemas operativos hasta compiladores y sistemas de bases de datos. Una de sus más relevantes cualidades es la redefinición de operadores y palabras claves para manejar las clases. Algunas de sus extensiones tienen aplicación fuera del contexto de la programación orientada a objetos (C++ con clase , 2007).

¿Por qué C++ en su estándar ANSI 98?

Se escoge este lenguaje pues, es el lenguaje en el que se está desarrollando el sistema de Video Vigilancia Xilema Suria y la solución a crear es una contribución a dicho proyecto. El lenguaje C++ es multiplataforma lo que brinda la posibilidad de desarrollar software con él sin la necesidad de pagar licencia. Además, de ser un lenguaje multinivel, es versátil y posee las ventajas de C en cuanto a flexibilidad, eficiencia y concisión, lo cual será muy bien aprovechado en el desarrollo del *plugin* para integrar la aplicación al sistema de Video Vigilancia Xilema Suria.

Java

En su libro (Downey, 2012) menciona que según la compañía *Sun Microsystems* (creadores del lenguaje Java) este es un lenguaje que puede ser muy bien aprovechado por ser simple, orientado a objetos y porque cuenta con amplia distribución. Java es muy robusto y seguro, con una arquitectura versátil. Entre sus grandes ventajas se puede mencionar que es portable, de altas prestaciones, multitarea y dinámico.

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

En la presente investigación se hará uso específicamente de la ventaja de este lenguaje en cuanto a que es multiplataforma y orientado a objetos.

¿Por qué utilizar Java?

En la investigación en curso se hace uso del lenguaje de programación java, pues se cuenta con una biblioteca (*Sphinx*) que está implementada en dicho lenguaje y es necesario su aprovechamiento para la implementación del sistema. Esta biblioteca ayuda en gran medida al reconocimiento de voz a partir del audio captado a través de un flujo de video.

1.6.7. Entorno de Desarrollo Integrado

Qt Creator 2.4.1

Qt Creator es un entorno completo de desarrollo integrado (IDE) para la creación de aplicaciones con el *framework* Qt. Qt fue creado para el desarrollo de aplicaciones e interfaces de usuario, con el propósito de desplegarlas a múltiples sistemas operativos tanto móviles como de escritorio. Su objetivo principal es el conocimiento de la necesidad del programador que busca fundamentalmente, facilidad de uso (*Qt Project Hosting SA*, 2013).

¿Por qué usar Qt Creator 2.4.1?

Se decide usar *Qt Creator* en la presente investigación debido a que permite desarrollar aplicaciones de forma rápida. Una de las ventajas más valiosas que posee es que permite al equipo de desarrollo compartir el proyecto en disímiles plataformas, ya sea *Microsoft Windows*, *Linux* o *Mackintosh*. *Qt Creator* posibilita además desarrollar aplicaciones en C++, características que serán aprovechadas en el desarrollo de la aplicación para el reconocimiento de palabras claves en audio. Se emplea además este IDE pues la mayoría de los proyectos del departamento del centro GEYSED lo emplean en el desarrollo de sus aplicaciones y como la presente es una aplicación para contribuir al sistema de Video Vigilancia Xilema Suria, que está implementado utilizando *Qt Creator*, entonces se deriva que esta aplicación también lo esté.

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

Eclipse 3.7.2

Entorno de desarrollo integrado de código abierto y multiplataforma lo permite desarrollar software con él sin la necesidad de pagar licencia. Este es un *framework* multi-nivel en el que su trabajo con *plugins* permite integrar diversos lenguajes sobre un mismo IDE. Se le pueden integrar otras aplicaciones accesorios que pueden resultar útiles durante el proceso de desarrollo (Robledo Sacristán, y otros, 2012).

¿Por qué usar Eclipse 3.7.2?

En el desarrollo de la investigación en curso se ha decidido el empleo de Eclipse como entorno de desarrollo integrado por contar con las características requeridas para el desarrollo de la aplicación que se quiere obtener. Se destaca en ellas que en este entorno de trabajo se puede usar java como lenguaje de programación que se ha escogido con anterioridad y que es fundamentalmente de código abierto y multiplataforma.

1.6.8. Framework de Desarrollo

Qt 4.8.0

Qt es un *framework* de desarrollo de aplicaciones multiplataforma. Viene acompañado de un conjunto de herramientas para facilitar su uso. Incluye una serie de funcionalidades que no están normalmente en C++. Para ello utiliza su propio preprocesador llamado MOC (*Meta Object Compiler*) (*qt-project.org*, 2012). Qt presenta las características necesarias para dar cumplimiento al objetivo general trazado, ya que está diseñado para desarrollar aplicaciones e interfaces de usuario y desplegarlas a múltiples sistemas operativos, tanto móviles como de escritorio.

¿Por qué usar Qt 4.8.0?

Se considera que para implementar la solución de la presente investigación se hará uso del *framework* Qt en su versión 4.8.0. Esto se debe fundamentalmente a que es el *framework* que se utiliza en la mayoría de los proyectos del departamento GEYSED para desarrollar software. Otra razón es la característica de ser multiplataforma y de código abierto. Esto facilita que la futura aplicación pueda adaptarse tanto para el sistema operativo Linux, *Windows* o *Mackintosh*, permitiendo que una aplicación pueda ser compilada y utilizada en cualquier plataforma.

Capítulo 1: Fundamentación Teórica

Asociada al Reconocimiento del Habla

Eclipse 3.7.2

Desarrollado originalmente por IBM (International Business Machines), es un conjunto de herramientas de programación de código abierto y multiplataforma para el desarrollo de aplicaciones cliente. Esta herramienta ha pasado a tener su propia compañía (*Eclipse Foundation*) y es usada para la creación de entornos de desarrollo como java. Eclipse tiene como objetivo la producción de un marco ingenieril de procesos de software personalizados con un contenido de ejemplos de procesos. Sus herramientas apoyan a varios tipos de proyectos y estilos de desarrollo (*Eclipse Foundation, 2014*)

¿Por qué usar Eclipse 3.7.2?

Para el empleo de Eclipse como *framework* de desarrollo se ha considerado que este es extensible, además de que incluye su entorno de desarrollo integrado. Eclipse es abierto y multiplataforma lo que será muy favorecedor en el desarrollo de la aplicación para evitar cambiar código cuando sea compilada en uno u otro sistema operativo.

1.1. Conclusiones Parciales

En este capítulo fueron abordados una serie de conceptos de marcada importancia para el desarrollo de la investigación. Esto permitió al equipo de trabajo poseer un mayor dominio del problema existente en el sistema de Video Vigilancia Xilema Suria, asociado al reconocimiento de palabras claves en audio. El estudio de un conjunto de herramientas para el reconocimiento del habla, permitió identificar las principales técnicas asociadas a dicho proceso y seleccionar según sus características la adecuada para la construcción del sistema. Se caracterizaron las herramientas y tecnologías a utilizar en el desarrollo de la aplicación enunciándose de cada una de ellas las principales características y las facilidades que ofrecen al equipo de desarrollo, favoreciendo la posible integración con sistemas que lo soporten y garantizando futuras actualizaciones del sistema. Una vez planteado lo descrito en el capítulo quedan sentadas las bases para el desarrollo de la aplicación garantizando la facilidad por parte de los programadores a la hora de codificar el sistema en desarrollo y la comprensión por parte de los usuarios finales.

Capítulo 2: Análisis y Diseño

2.1. Introducción

En este capítulo se abordarán los temas relacionados con el análisis y diseño de la solución. Inicialmente se especifican los requisitos funcionales y no funcionales que debe cumplir el sistema, así como el modelo de dominio y los casos de usos con las descripciones correspondientes de cada uno. También se definen las clases de análisis y diseño de los casos de usos. Se presenta la arquitectura del sistema, describiendo además las clases del flujo de trabajo análisis y diseño.

2.2. Modelo de Dominio

Los procesos que se modelan en el flujo de trabajo actual no complementan la realización de un modelo de negocio, ya que no están bien definidos con una secuencia de eventos bien establecida. Es por ello que se crea un modelo de dominio con el objetivo de sentar las bases para la captura de los objetos más importantes y para los eventos que se desarrollan en el entorno del sistema.

2.2.1. Descripción del Modelo de Dominio

El Gestor del Sistema de Video Vigilancia Suria cuenta con varios módulos, uno de ellos es el de Análisis el cual posee varios video sensores de los que recibe la información que estos detectan. Un video sensor es el proceso que se realiza sobre el flujo de video captado de una cámara de video vigilancia y mediante el que puede ser detectada una acción.

Capítulo 2: Análisis y Diseño

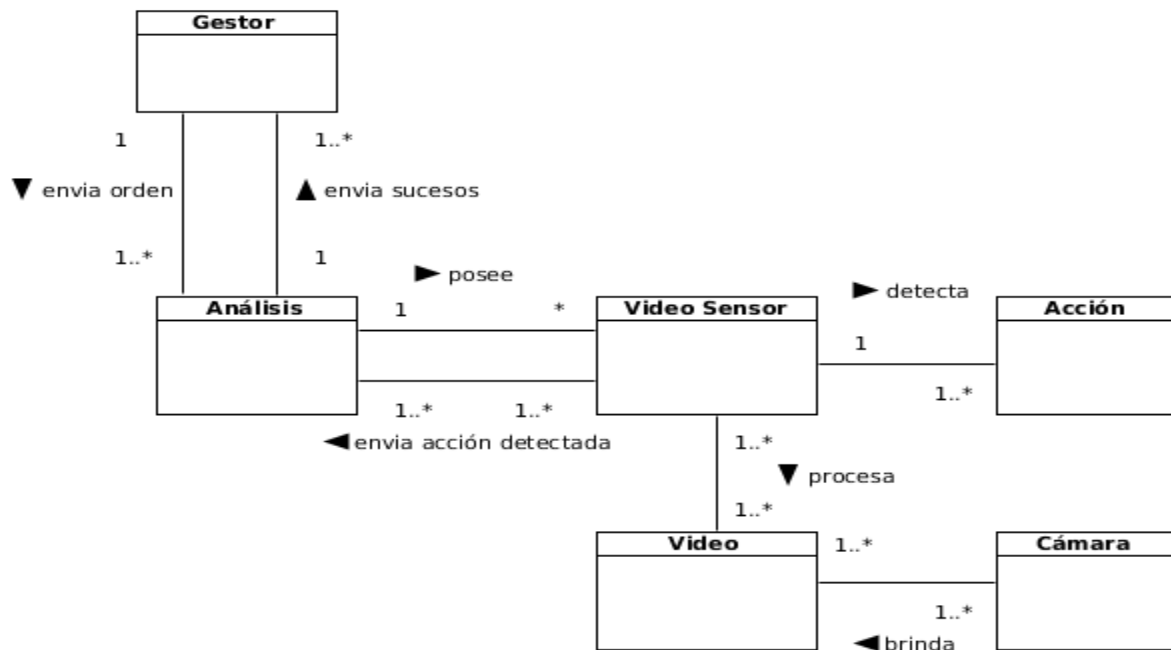


Fig. 1 Modelo de Dominio

2.2.2. Conceptos Asociados al Modelo de Dominio

Gestor: Módulo encargado de orquestar toda la comunicación entre los diferentes módulos del sistema de Video Vigilancia Xilema Suria.

Análisis: Es uno de los módulos que componen el Gestor del sistema de Video Vigilancia Xilema Suria.

Video: Secuencia de imágenes captada por la cámara.

Video Sensor: Algoritmo que permite el procesamiento sobre un flujo de video para identificar variaciones significativas en la secuencia de imágenes.

Cámara: Dispositivo de captura de video.

Acción: Actividad realizada en el flujo de video.

Capítulo 2: Análisis y Diseño

2.3. Requisitos Funcionales (RF)

Los requisitos funcionales son las aclaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y como se debe comportar. En algunos casos, los requisitos funcionales del sistema deben aclarar explícitamente lo que el sistema no debe hacer (Sommerville, 2005).

En la presente investigación se definen los siguientes requisitos funcionales:

Tabla. 1 Requisitos Funcionales

No.	Requisitos	Descripción
RF1	Obtener flujo de video de las cámaras de video vigilancia.	El sistema debe ser capaz de obtener el flujo de video de las cámaras de video vigilancia Xilema Suria.
RF2	Extraer audio del flujo de video.	El sistema debe extraer el audio presente en el flujo de video que ha sido previamente captado.
RF3	Transcribir audio	El sistema debe, transformar a texto la voz presente en el flujo de audio.
RF4	Notificar al Módulo de Análisis	El sistema debe, una vez analizado el audio, notificar al Módulo de Análisis enviándole el flujo de texto reconocido.

2.4. Requisitos no Funcionales (RNF)

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema (Sommerville, 2005).

Capítulo 2: Análisis y Diseño

2.4.1. RNF1 Software

- La aplicación debe funcionar correctamente en el sistema operativo Windows a partir de su versión Windows7.
- La aplicación debe funcionar correctamente en el sistema operativo GNU/Linux a partir de su versión 12.04.

2.4.2. RNF2 Hardware

- Procesador Intel Core 2 Duo o Dual Core de 2 GHz o superior.
- 512 MG de memoria RAM o superior.
- Espacio en disco duro de 15 GB o superior.

2.4.3. RNF3 Rendimiento

- El sistema deberá tener un tiempo máximo de respuesta de 10 segundos para mostrar la palabra luego de que esta sea pronunciada.
- El sistema deberá reconocer de manera exacta la mayor cantidad de palabras posibles.

2.5. Definición de Casos de Uso

2.5.1. Actor del Sistema

Tabla. 2 Actor del Sistema

Actor	Descripción
Módulo de Análisis	El módulo de Análisis del sistema de Video Vigilancia Xilema Suria es quien le envía la orden al reconocedor para que este comience a procesar el flujo de video.

Capítulo 2: Análisis y Diseño

2.5.2. Diagrama de Caso de Uso del Sistema

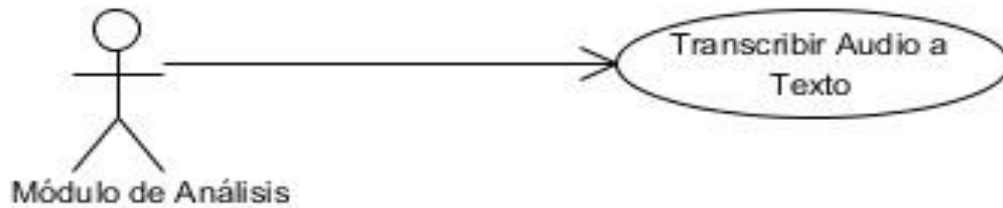


Fig. 2 Diagrama de Caso de Uso del Sistema

2.5.3. Descripción del Caso de Uso del Sistema

A continuación se muestra la descripción del caso de uso involucrado en el sistema.

Tabla. 3 Descripción del Caso de Uso: Transcribir de Audio a Texto

Nombre	Transcribir Audio a Texto.
Objetivo	Obtener el flujo de video de las cámaras de video vigilancia del sistema Xilema Suria y que este sea transcrito a texto.
Actores	Módulo de Análisis.
Resumen	El caso de uso inicia cuando el actor envía la orden para comenzar a procesar el flujo de video y concluye cuando es emitido el flujo de texto transcrito.
Complejidad	Alta
Prioridad	Crítico.
Precondiciones	Para obtener un flujo de video y transcribirlo a texto, las cámaras deben estar transmitiendo.
Postcondiciones	Se ha obtenido el flujo de texto que se ha de notificar al Módulo de Análisis.

Capítulo 2: Análisis y Diseño

Flujo de eventos		
Flujo básico: Obtener Flujo de Video		
No.	Actor	Sistema
1.	Envía una orden al reconocedor para que comience a procesar el flujo de video.	
2.		Obtiene el flujo de video de las cámaras de video vigilancia.
3.		Extrae el audio del flujo de video.
4.		Transcribe de audio a texto.
5.		Notifica al Módulo de Análisis con el flujo de texto transcrito.
6.		Termina el caso de uso
Flujo Alternativo		
2: Procesar flujo de video.		
No	Actor	Sistema
2.a		Las cámaras del sistema no están funcionando y por tanto se muestra una señal de alerta indicando que no se ha encontrado el flujo de video.
Relaciones	CU Extendidos	No tiene
	CU Incluidos	No tiene
RF	RF1, RF2, RF3, RF4	

Asuntos Pendientes	No procede.
---------------------------	-------------

2.6. Arquitectura de Software

La arquitectura de software según (Pressman, 1997) es la representación capaz de capacitar al ingeniero del software con el objetivo de que este realice un análisis consecuente con los requisitos descritos y considere las alternativas arquitectónicas en una etapa en la cual, es relativamente fácil hacer cambios asociados a la construcción del software.

Otros autores (Jacobson, y otros, 2000) definen la arquitectura de software como el elemento que especifica el arquitecto del software que permite controlar el desarrollo del sistema desde una perspectiva técnica. La arquitectura de software se centra tanto en los elementos estructurales significativos del sistema, como subsistemas, clases, componentes y nodos, como en las colaboraciones que tienen lugar entre estos elementos a través de las interfaces.

Partiendo de lo antes expuesto se puede hacer mención a que es necesaria la arquitectura de software para tener un claro entendimiento del sistema, organizar el desarrollo, fomentar la reutilización del sistema y hacerlo evolucionar. La arquitectura de software guiada por los casos de uso garantiza que el sistema proporcione las funcionalidades y usos deseados.

2.6.1. Estilo Arquitectónico

Los estilos arquitectónicos son los encargados de hacer una detallada descripción de una categoría del sistema, pudiendo ser ésta: un conjunto de componentes que realizan una determinada función para el sistema, un conjunto de conectores para posibilitar la comunicación entre los componentes, restricciones que definen como se pueden integrar los componentes que forman el sistema y modelos semánticos que permiten al diseñador entender las propiedades globales de un sistema para analizar las propiedades conocidas de sus partes constituyentes (Pressman, 1997).

Para el desarrollo de la investigación en curso se emplea como estilo arquitectónico Flujo de Datos. Partiendo de que, lo que se ha de analizar y procesar es el flujo de video proveniente de una cámara de video vigilancia, se hace uso de este estilo arquitectónico pues dicho flujo de video en su paso por el

Capítulo 2: Análisis y Diseño

sistema a construir será transformado y manipulado para obtener de él un texto que reproduzca lo que se ha hablado en el mismo.

2.6.2. Patrón Arquitectónico

Al emplear Flujo de Datos se deriva la utilización del patrón o sub-estilo arquitectónico de tuberías y filtros el cual según (Pressman, 1997) contiene un grupo de componentes llamados filtros conectados por tuberías que transmiten datos de un componente al siguiente. El trabajo de uno de estos filtros es independiente de su vecino y además de los componentes de entrada y de salida. Por lo explicado con anterioridad al captar el video de una cámara de video vigilancia se necesita que el procesado de este video se haga de forma secuencial y con procesos independientes. Para el desarrollo de la presente investigación el proceso de reconocimiento de palabras claves en audio se hará empleando las clases implementadas como filtros que transforman el flujo de video en su paso por ellas y las tuberías serán la relación entre las mismas.

2.6.3. Patrones de Diseño

Patrones de software para la asignación de responsabilidades o *General Responsibility Assignment Software Patterns (GRASP)* que son utilizados en el desarrollo del sistema.

➤ Experto

Este patrón está diseñado para que la responsabilidad de realizar una función determinada sea de la clase que tiene o puede tener la información requerida. Está presente en la clase *CC_hilo_capturar_flujo* de manera que la responsabilidad de *CapturarFlujo* está asignada a la clase que contiene toda la información necesaria para ello.

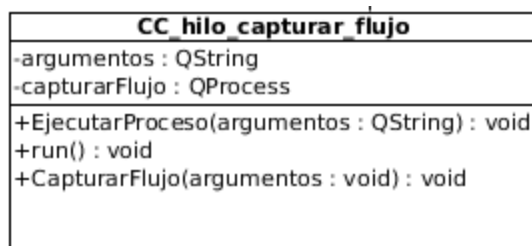


Fig. 3 Evidencia de patrón Experto

Capítulo 2: Análisis y Diseño

➤ Creador

Está diseñado para asignar responsabilidades relacionadas con la creación de objetos. Su intención es encontrar un creador que necesite conectarse al objeto creado en alguna situación. Está presente en la clase *CC_Reconocedor* que es la encargada de crear instancias de las clases *CC_hilo_capturar_flujo* y *CC_hilo_reconocer*.

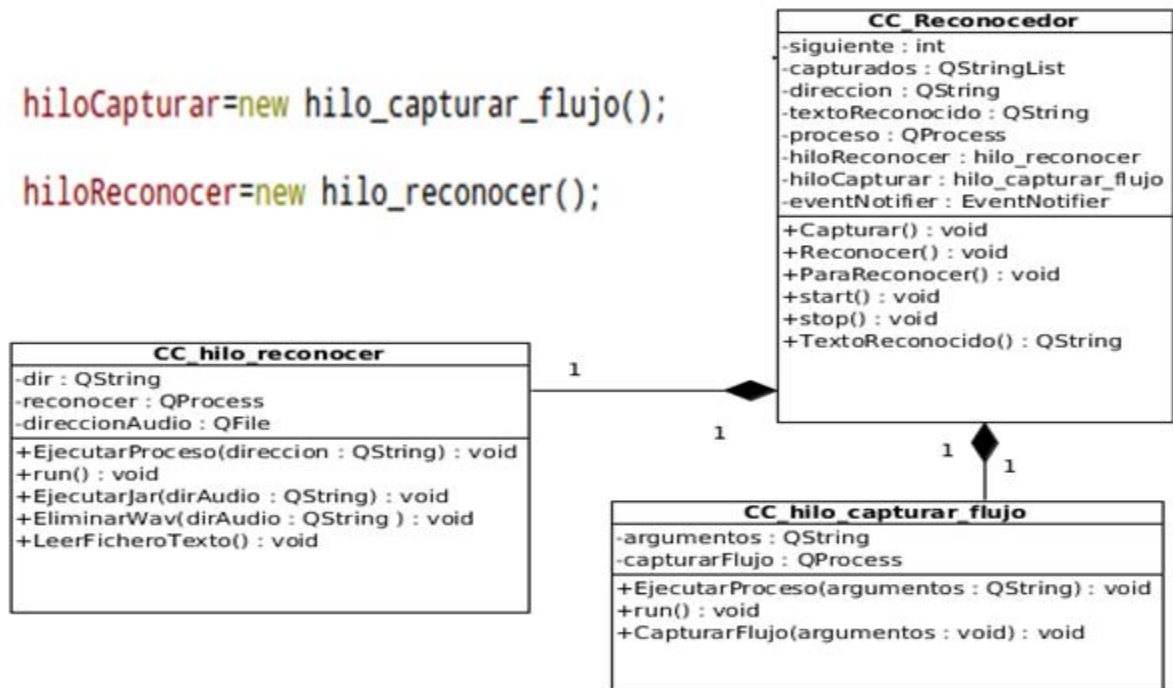


Fig. 4 Evidencia del patrón Creador

➤ Controlador

Está diseñado para asignar la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Esto facilita la centralización de actividades y se encuentra presente en la clase *CC_Reconocedor*, que es la encargada de atender los eventos del sistema que son generados por un actor externo y se asocian a operaciones del sistema como *Capturar*, *Reconocer* y *TextoReconocido*.

Capítulo 2: Análisis y Diseño

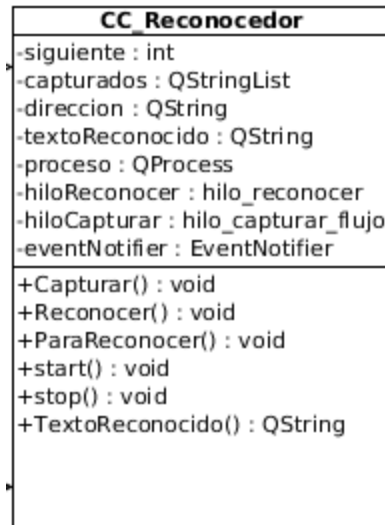


Fig. 5 Evidencia del patrón Controlador

➤ Alta Cohesión

Se encarga de que las clases del diseño cumplan con las funcionalidades que definen y que estas estén interrelacionadas. Su aplicación se evidencia en la clase *CC_hilo_capturar_flujo* ya que en esta, las funcionalidades de *EjecutarProceso* y *CapturarFlujo*, están altamente relacionadas entre sí, pues al ejecutarse el proceso definido se comienza a capturar el flujo de video transmitido por la cámara de video vigilancia y en un momento determinado puede ser detenida dicha captura.

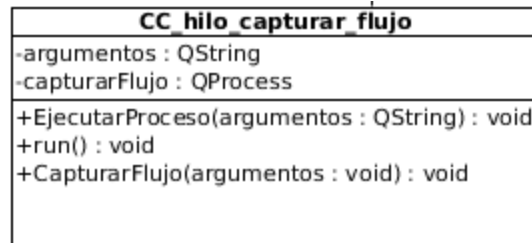


Fig. 6 Evidencia del patrón Alta Cohesión

➤ Bajo Acoplamiento

Se encarga de que las clases del diseño colaboren unas con otras, siempre y cuando esta colaboración se mantenga en un mínimo aceptable, reduciendo el impacto de posibles cambios en el sistema. Su evidencia se muestra en la clase *CC_Reconocedor* pues ella mantiene un bajo acoplamiento con las

Capítulo 2: Análisis y Diseño

clases *CC_hilo_capturar_flujo*, *CC_hilo_reconocer* y *CC_EventNotifier* de manera que invoca los servicios de un objeto del tipo de alguna de estas dos clases.

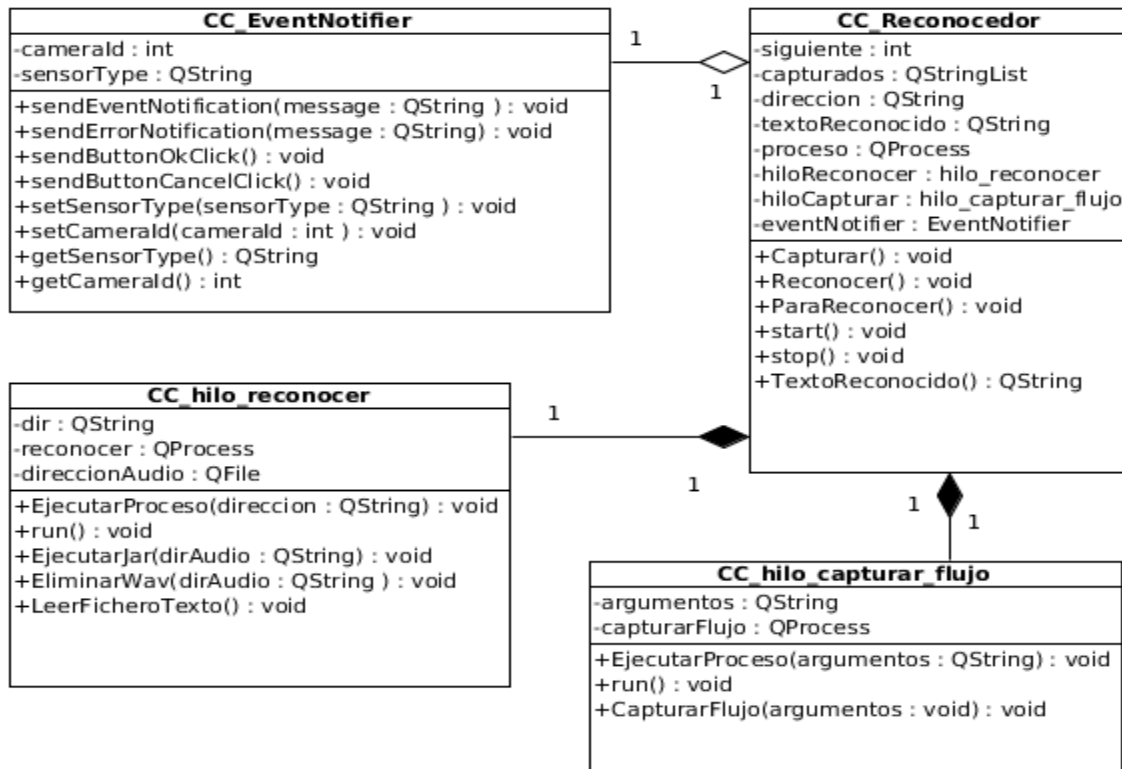


Fig. 7 Evidencia del patrón de Bajo Acoplamiento

Patrones Gang of Four (GoF)

➤ Interfaz

Interfaz es otro patrón utilizado para el diseño del sistema y se pone de manifiesto en la clase *ISensorPlugin*, ya que la misma es una clase virtual pura debido a su implementación en el lenguaje de programación C++, que solamente se encarga de definir las operaciones que deben ser implementadas en la clase *CC_Reconocedor*. Todas las clases que quieran utilizar este comportamiento deberán implementar dicha interfaz.

Capítulo 2: Análisis y Diseño

```
class ISensorPlugin : public QObject
{
    Q_OBJECT

public:
    ISensorPlugin(){}

    virtual void start()=0;
    virtual void stop()=0;
    virtual QWidget* getConfigInterface()=0;
    virtual QHash<QString,QString> getParameters()=0;
    virtual void setParameters(QHash<QString,QString> parameters)=0;
    virtual QString getSensorType()=0;
    virtual EventNotifier* getNotifier()=0;
};
```

Fig. 8 Evidencia del patrón interfaz

2.7. Diseño de Software

Diseñar software es la parte fundamental de la ingeniería de software. Ello facilita en gran medida, la creación de un modelo que funciona como esquema para la implementación. Es en este momento del desarrollo del software donde quedan definidos artefactos como: clases, subsistemas e interfaces, además de las relaciones que existen entre ellos.

2.7.1. Diagrama de Clases del Diseño

Los diagramas de clases del diseño son una forma de asignación de responsabilidades planteada inicialmente. Son utilizados para visualizar la comunicación entre las clases involucradas en el sistema. Es la creación de la primera imagen ingenieril de cómo quedará la implementación del prototipo funcional (Jacobson, y otros, 2000). A continuación se muestran los diagramas de clases del diseño correspondiente al sistema a construir.

Capítulo 2: Análisis y Diseño

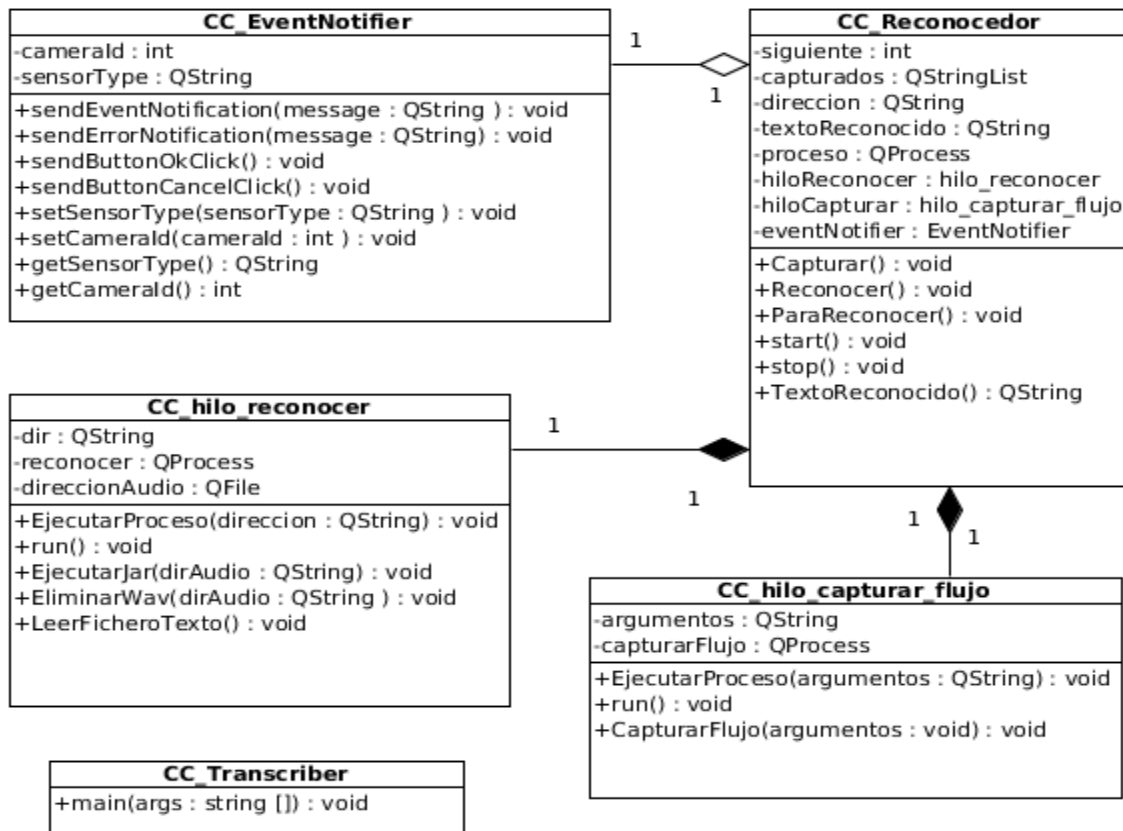


Fig. 9 Diagrama de Clases del Diseño

Tabla. 4 Descripción de las Clases del Diseño

Nombre de la Clase	Tipo de Clase	Descripción
CC_Reconocedor	Controladora	Se encarga de la captura y procesado del flujo de video que llega en tiempo real y de la comunicación con el módulo de Análisis.
CC_hilo_capturar_flujo	Controladora	Es la clase especializada en capturar el flujo de video transmitido por la cámara de video vigilancia.
CC_hilo_reconocer	Controladora	Es la clase especializada en reconocer palabras en el flujo de audio capturado de la cámara de video vigilancia.

Capítulo 2: Análisis y Diseño

CC_Transcriber	Controladora	Se encarga de procesar un audio para obtener el texto presente en él.
CC_EventNotifier	Controladora	Es la clase encargada de notificar al gestor del Sistema de Video Vigilancia Xilema Suria la ocurrencia de algún evento significativo ocurrido en el mismo.

2.8. Diagramas de Interacción

Los diagramas de interacción son una forma de ilustrar como los objetos interactúan entre sí por medio de mensajes. El nombre mensaje indica aquí la operación que realiza un objeto al ser invocado, o que realiza una interfaz que proporciona el objeto. Su importancia radica en que pueden modelar aspectos dinámicos de un sistema. (Alfonso Amo, y otros, 2005)

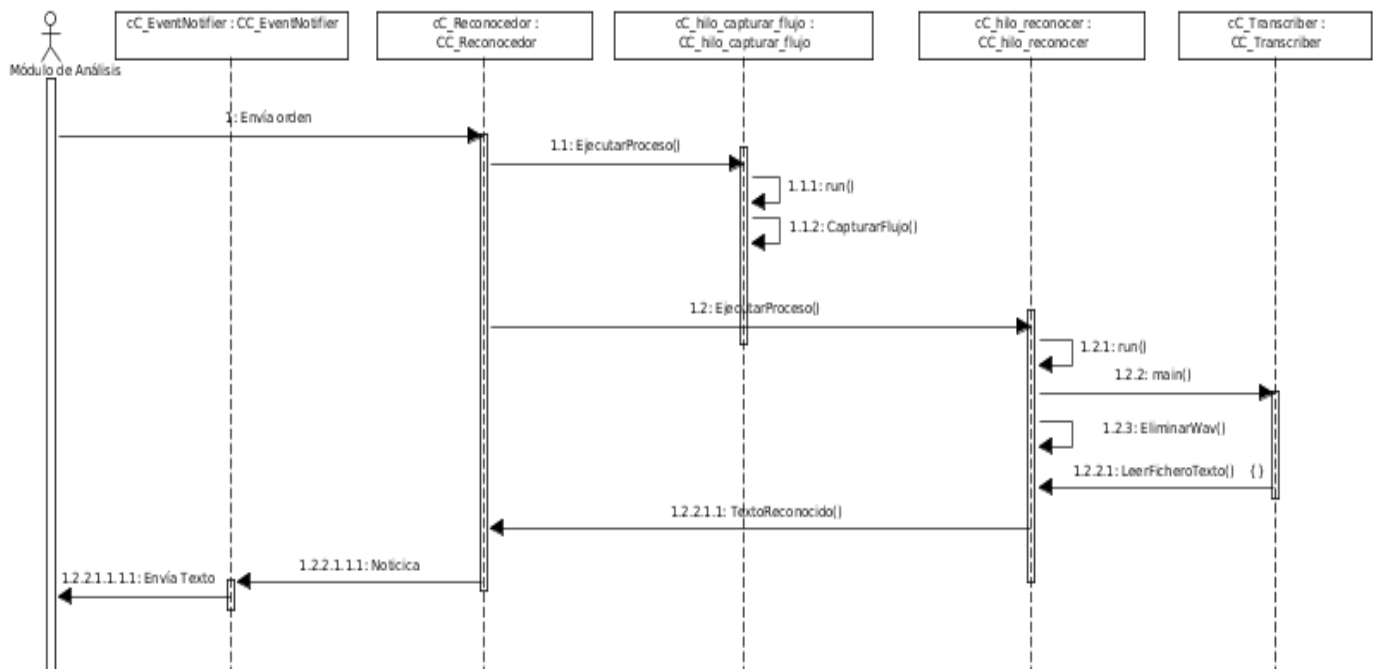


Fig. 10 Diagrama de Secuencia Genérico

2.9. Conclusiones Parciales

En este capítulo se trataron temas referentes al análisis y diseño del video sensor para el reconocimiento de palabras claves en audio, lo que permitió arribar a las siguientes conclusiones:

La modelación de las clases del dominio permitió establecer un punto de partida para lograr un correcto diseño del sistema, permitiendo además la obtención de un mejor razonamiento del problema a resolver mediante la presentación de un diagrama y los principales conceptos que intervienen en el dominio. Se estableció la línea base de la arquitectura que tendrá el sistema a desarrollar empleándose el patrón arquitectónico de tuberías y filtros como una especificación de la arquitectura flujo de datos. Se identificaron los patrones de diseño GRASP y GoF utilizados en el desarrollo de la aplicación, permitiendo facilitar la asignación de responsabilidades logrando un diseño de software que sirva de apoyo a la implementación del sistema. Con la realización del diagrama de clases del diseño se obtuvo una visión más exacta del sistema en términos de implementación, lo que ayudará al programador a ampliar su visión sobre lo que se desea implementar. La representación del diagrama de casos de uso y su respectiva descripción, así como la representación del diagrama de secuencia, permitieron al desarrollador tener una vista más clara de lo que se pretende hacer.

Capítulo 3: Implementación y Prueba

3.9. Introducción

Una vez concluido el análisis y diseño de la solución propuesta se da paso al siguiente flujo de trabajo. En el presente capítulo se realiza la implementación de las clases y objetos que componen la solución propuesta. Son generados los diagramas de despliegue y componentes para una mejor comprensión de la misma. Se da una descripción de los estilos que se emplearon en la codificación y una vez concluida la implementación se describen las pruebas que se le realizaron al sistema con el objetivo de corregir posibles errores detectados.

3.2. Modelo de Implementación

En el flujo de trabajo de implementación se analizan los resultados del diseño con el objetivo de implementar el sistema en términos de componentes. El propósito general de este es desarrollar la arquitectura y el sistema como un todo. Aquí se planifican las integraciones necesarias del sistema para cada iteración y se distribuye el mismo asignando componentes ejecutables a nodos en el diagrama de despliegue. Se implementan clases y subsistemas encontrados y se realizan las pruebas a cada componente (Jacobson, y otros, 2000).

3.2.1. *Implementación de Software*

La implementación es el centro durante las iteraciones de construcción. Es el proceso que asegura la operatividad del sistema a partir de lo cual el usuario obtiene sus beneficios. La fase de implementación es donde se define la organización y estructura del código en términos de subsistemas estructurados, se definen las clases y objetivos en términos de componentes.

3.3. Diagrama de Despliegue

Los diagramas de despliegue permiten mostrar la arquitectura del sistema en tiempo de ejecución. Es empleado tanto para el diseño como para la implementación del software. En ellos se pueden distinguir componentes y nodos que ejecutan dichos componentes, así como relaciones entre ellos. (Falgueras, 2002)

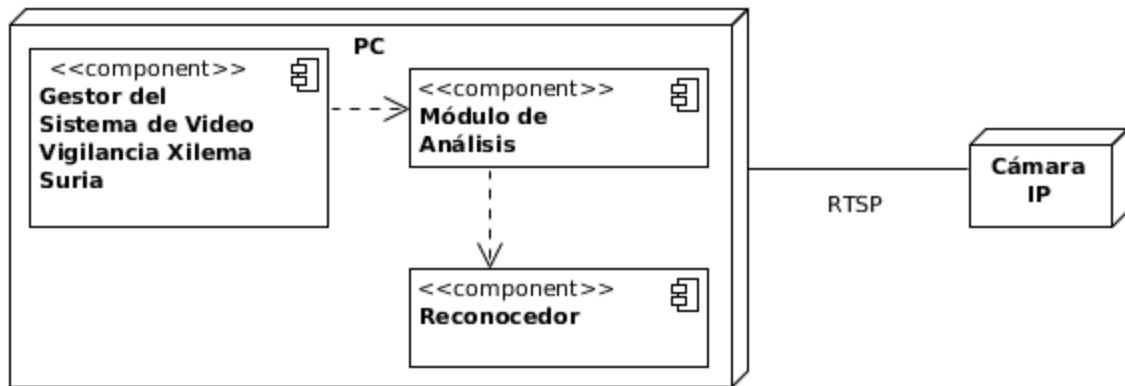


Fig. 11 Diagrama de Despliegue

3.4. Diagrama de Componentes

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en un modelo de diseño. Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean estos componentes de código fuente, binarios o ejecutables. (Rumbaugh, y otros, 2000)

A continuación se representa el diagrama de componentes de código fuente, del Video sensor para el reconocimiento de palabras claves en audio.

Capítulo 3: Implementación y Prueba

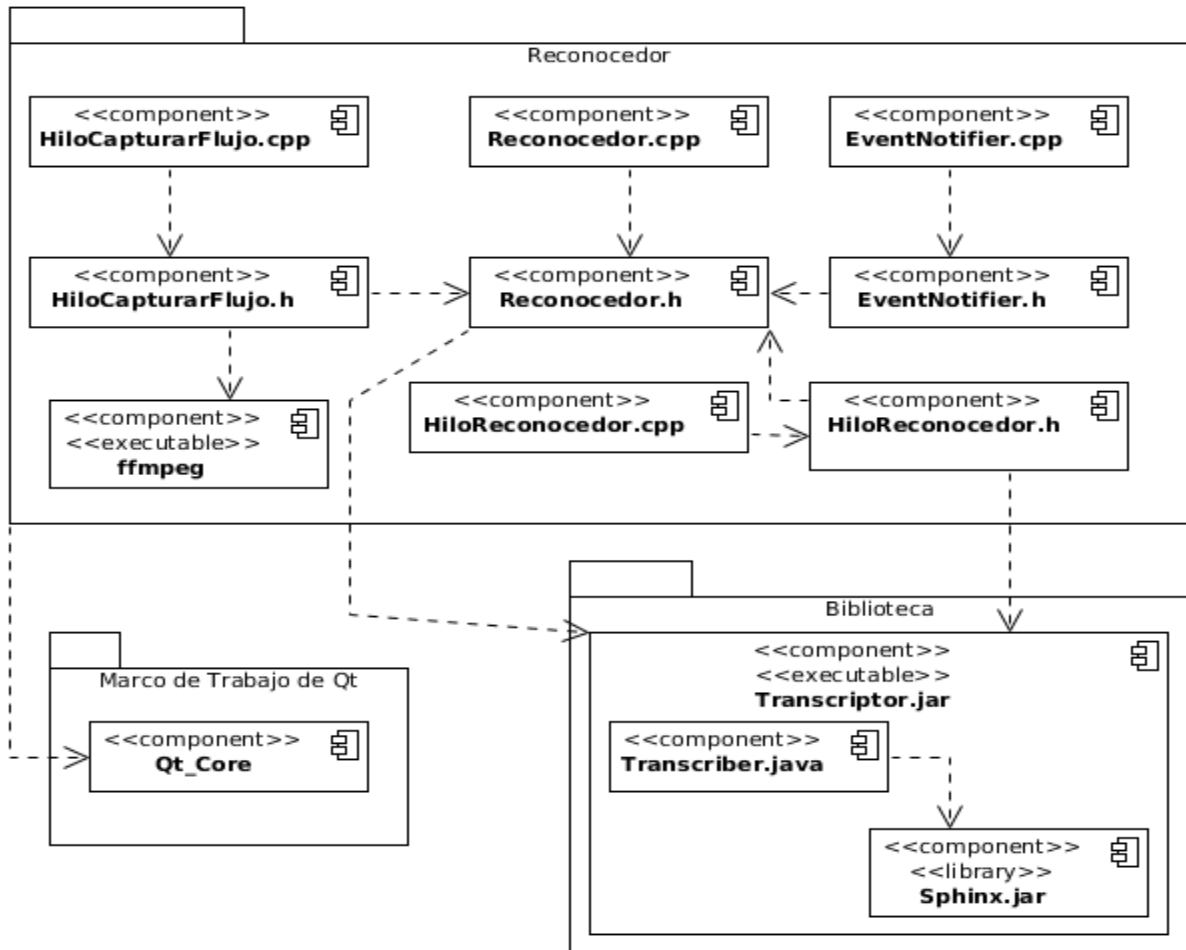


Fig. 12 Diagrama de Componentes

3.5. Estándares de Codificación

Los estándares de codificación implican todos los aspectos de la generación de código. El propósito general de estos es que los proyectos posean una arquitectura y un estilo consistente que facilite el entendimiento del sistema y su mantenimiento. Mediante un estándar de codificación se debe dejar clara la intención del código y permitir a los programadores acercarse al lenguaje natural.

3.5.1. Estilo de Codificación Utilizado

Los estándares de codificación definidos para la presente investigación son los siguientes:

Capítulo 3: Implementación y Prueba

- Declarar cada variable en una línea independiente.
- Los nombre de las variables que sean cortos, significativos y con letra inicial minúscula.
- Se añaden comentarios solo de tipos aclarativos, solo cuando sea necesario.
- Los nombres de los métodos comienzan con letra inicial mayúscula y si hay más de una palabra entonces ésta comienza también con mayúscula.
- La “{” no debe aparecer en la misma línea de la declaración del código.
- La elección de un nombre de variable debe ser diseñado para demostrar el propósito de su uso a cualquier observador.
- Evitar el uso de un solo carácter en los nombres de las variables.
- Las variables deben ser del tipo “prefijo” “nombre” donde el prefijo es el tipo de dato de la variable y el prefijo describe la variable.

3.6. Pruebas

Luego de generado el código fuente del software este debe ser probado para discutir y (en su medida) corregir la mayor cantidad de errores posibles. Las pruebas de software son un elemento fundamental para la garantía de calidad del mismo. Las posibilidades de fallos humanos en la creación de software son enormes y es por ello la necesidad de crear pruebas minuciosas y bien planificadas. El objetivo fundamental de este proceso es diseñar casos de prueba que tengan una alta probabilidad de encontrar errores y para ello se emplean técnicas que faciliten una guía para probar la lógica interna de los componentes de software, las funcionalidades y el comportamiento y rendimiento del mismo.

3.6.1. Pruebas de Caja Blanca

La prueba de caja blanca es un método basado en el diseño de casos de pruebas que usa la estructura de control del diseño procedimental para obtener los casos de prueba. A través de estos algoritmos el ingeniero de software puede obtener casos de prueba que garanticen que se ejercitan por los menos una vez todos los caminos independientemente de cada módulo. Que se ejerciten además todas las soluciones lógicas en sus vertientes verdadera y falsa, que se ejecuten todos los bucles en sus límites operacionales, y que ejerciten las estructuras internas de datos para asegurar su validez (Pressman, 1997).

Camino Básico

Capítulo 3: Implementación y Prueba

Camino básico es una técnica de prueba de Caja Blanca que consiste en derivar casos de prueba a partir de un conjunto de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el grafo de flujo asociado al código y se calcula su complejidad ciclomática. Se identifican los nodos y según las instrucciones serán las aristas que conecten a esos nodos que son los que permiten dibujar el grafo de flujo. Se selecciona el método de camino básico como prueba de caja blanca ya que este permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizarán que durante la prueba se ejecute por lo menos una vez cada sentencia del programa. (Pressman, 1997)

El procedimiento para llevar a cabo la técnica de prueba del camino básico es el siguiente:

1. A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
2. Se calcula la complejidad ciclomática del grafo de flujo.
3. Se determina el conjunto básico de caminos independientes.
4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Cálculo de la Complejidad Ciclométrica ($V(G)$)

La complejidad ciclométrica de un grafo de flujo se puede calcular de 3 formas diferentes:

$V(G)$ = número de regiones.

$V(G) = E - N + 2$; donde E es el número de aristas del y N el número de nodos del grafo de flujo.

$V(G) = P + 1$; donde P es el número de nodos predicados incluidos en el grafo de flujo. Un nodo predicado es aquel que representa una condición, por lo que de él salen dos o más caminos.

Casos de Prueba

Los Casos de pruebas definen un conjunto específico de entradas de pruebas, ejecución de condiciones y resultados esperados. Luego de la elaboración de los Grafos de Flujos y los caminos a recorrer, se preparan los casos de prueba que forzarán la ejecución de cada uno de esos caminos. Se escogen los datos de forma que las condiciones de los nodos predicados estén adecuadamente establecidas, con el fin de comprobar cada camino.

Capítulo 3: Implementación y Prueba

Funcionalidad Capturar

```
void Reconocedor::Capturar()
{
    QString argumentos;
    argumentos="-i rtsp://root:Admin.123@10.54.13.250:554/live.sdp "
              +QApplication::applicationDirPath()+"/Audios/audio"+QString::number(siguiete)+".wav";
    capturados.append(QApplication::applicationDirPath()+"/Audios/audio"
                    +QString::number(siguiete)+".wav");
    if (bandera) {
        hiloCapturar->quit();
        hiloCapturar->EjecutarProceso(argumentos);
        siguiete++;
    }else{
        hiloCapturar->EjecutarProceso(argumentos);
        siguiete++;
        bandera=true;
    }
}
```

Fig. 13 Funcionalidad Capturar

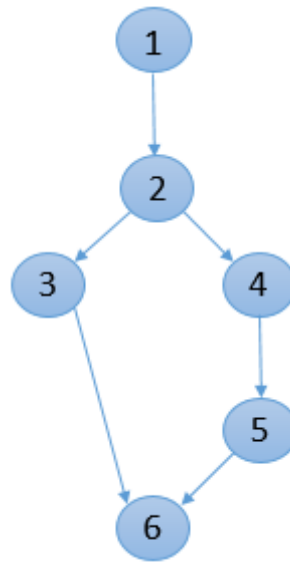


Fig. 14 Grafo de flujo de la funcionalidad Capturar

Capítulo 3: Implementación y Prueba

Tabla. 5 Caminos Básicos de la funcionalidad Capturar

No	Caminos Básicos
1	1-2-3-6
2	1-2-4-5-6

Una vez obtenidos los caminos básicos asociados al Grafo de Flujo de la funcionalidad en cuestión se procede a la creación de casos de pruebas por cada camino. A continuación se muestra el caso de prueba asociado a los caminos 1-2-3-6 y 1-2-4-5-6.

Tabla. 6 Caso de Prueba No1 de la funcionalidad Capturar

Caso de prueba para el camino básico No.1	
Camino	1-2-3-6
Descripción: una vez captado el flujo de video de la cámara, se extrae el audio y es guardada en una lista la dirección donde el mismo fue creado. Posteriormente se comprueba si el proceso de captura se está realizando, en caso afirmativo se elimina y se crea otro, evitando ejecutar dicho proceso más de una vez en el mismo flujo de video.	
Entrada:	Flujo de video captado por la cámara.
Resultado Esperado:	Flujo de audio captado por la cámara.
Resultado Obtenido:	Flujo de audio captado por la cámara.

Capítulo 3: Implementación y Prueba

Tabla. 7 Caso de Prueba No2 de la funcionalidad Capturar

Caso de prueba para el camino básico No.2	
Camino	1-2-4-5-6
Descripción:	una vez captado el flujo de video de la cámara y extraído el audio de él guardando en una lista la dirección donde fue creado el mismo, se comprueba si el proceso de captura ya se hizo y si no se hizo es creado otro, garantizando que el flujo de video sea captado en su totalidad.
Entrada:	Flujo de video captado por la cámara.
Resultado Esperado:	Flujo de audio captado por la cámara.
Resultado Obtenido:	Flujo de audio captado por la cámara.

Funcionalidad Reconocer

```
void Reconocedor::Reconocer()
{
    QTimer demorar;
    demorar.start(); } 1
    while (demorar.elapsed()<8000) { 2
    }
    if (capturados.length()!=0) { 3
        this->direccion=capturados.at(proximo);
        hiloReconocer->EjecutarProceso(direccion);
        capturados.removeAt(0); } 4
    } 5
```

Fig. 15 Funcionalidad Reconocer

Capítulo 3: Implementación y Prueba

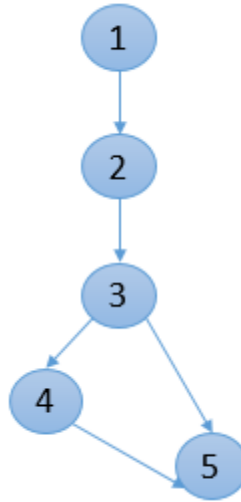


Fig. 16 Grafo de flujo de la funcionalidad Reconocer

Tabla. 8 Camino Básico de la funcionalidad Reconocer

No	Caminos Básicos
1	1-2-3-4-5
2	1-2-3-5

Una vez obtenidos los caminos básicos asociados al Grafo de Flujo de la funcionalidad Reconocer se procede a la creación de casos de pruebas por cada camino. A continuación se muestra el caso de prueba asociado a los caminos 1-2-3-4-5 y 1-2-3-5.

Tabla. 9 Caso de Prueba No1 de la funcionalidad Reconocer

Caso de prueba para el camino básico No.1	
Camino	1-2-3-4-5

Capítulo 3: Implementación y Prueba

Descripción: luego de un pequeño tiempo de espera para que el audio sea creado se comprueba si la lista de audios a analizar aún no está vacía y si es así entonces se continua con el proceso de reconocimiento.	
Entrada:	Flujo de audio captado por la cámara.
Resultado Esperado:	Flujo de texto transcrito del audio.
Resultado Obtenido:	Flujo de texto transcrito del audio.

Tabla. 10 Caso de Prueba No2 para la funcionalidad Reconocer

Caso de prueba para el camino básico No.2	
Camino	1-2-3-5
Descripción: luego de un pequeño tiempo de espera para que el audio sea creado se comprueba si la lista de audios a analizar aún no está vacía y si no finaliza el proceso de transcripción.	
Entrada:	Flujo de audio captado por la cámara.
Resultado Esperado:	Flujo de texto transcrito del audio.
Resultado Obtenido:	Flujo de texto transcrito del audio.

3.6.2. Pruebas de Rendimiento

Normalmente cuando se realizan pruebas de software, estas están diseñadas tanto para demostrar que el sistema satisface sus requerimientos como para descubrir problemas y defectos en el mismo. Las pruebas de rendimiento se han de diseñar con el objetivo de garantizar que el sistema pueda procesar la carga esperada en tiempo real y estresarlo realizando demandas que están fuera de los límites para los que el

Capítulo 3: Implementación y Prueba

software ha sido creado (Sommerville, 2005). El tema fundamental de dichas pruebas con respecto al reconocimiento de palabras claves en audio, es la transcripción de la mayor cantidad de palabras posibles que están siendo pronunciadas en un entorno de video vigilancia.

Al realizar un análisis de varios flujos de audio en distintos intervalos de tiempo captados en un ambiente controlado, estos arrojaron los siguientes resultados.

Tabla. 11 Palabras pronunciadas en el flujo de audio 1

Palabra pronunciada.	Cantidad de veces pronunciada.	Cantidad de veces reconocida.	Cantidad de veces reconocida de manera correcta.	Porcentaje de reconocimiento de cada palabra.	Porcentaje de exactitud de reconocimiento.
Asalto	24	23	20	95.83%	86.96%
Bomba	22	18	17	81.81%	94.44%
Peligro	27	22	20	81.48%	90.90%
Asaltaron	24	24	24	100%	100%
Pistola	17	13	12	76.47%	92.31%
Total porcentaje promedio de reconocimiento.				90.90%	93%

Una vez concluida la primera prueba haciendo un análisis de las palabras que el sistema es capaz de reconocer se debe tener en cuenta que estas tienen un porcentaje de reconocimiento de 90.90% con una exactitud de 93%. Esto representa un alto grado de reconocimiento en el ambiente en cuestión.

Capítulo 3: Implementación y Prueba

Tabla. 12 Palabras pronunciadas en el flujo de audio 2

Palabra pronunciada.	Cantidad de veces pronunciada.	Cantidad de veces reconocida.	Cantidad de veces reconocida de manera exacta.	Porcentaje de reconocimiento de cada palabra.	Porcentaje de exactitud de reconocimiento.
Asalto	24	19	17	79.17%	89.47%
Bomba	22	18	14	81.81%	77.77%
Peligro	27	22	19	81.48%	86.36%
Asaltaron	24	18	16	75%	88.88%
Pistola	17	10	9	58.82%	90%
Total porcentaje promedio de reconocimiento.				76.32%	86.21%

Concluidas las pruebas al segundo flujo de audio el porcentaje de reconocimiento y de exactitud ha decaído hasta 76.32% y 86.21% respectivamente. De ahí que sea necesario analizar un tercer flujo de audio.

Tabla. 13 Palabras pronunciadas en el flujo de audio 3

Palabra pronunciada.	Cantidad de veces pronunciada.	Cantidad de veces reconocida.	Cantidad de veces reconocida de manera exacta.	Porcentaje de reconocimiento de cada palabra.	Porcentaje de exactitud de reconocimiento.
----------------------	--------------------------------	-------------------------------	--	---	--

Capítulo 3: Implementación y Prueba

Asalto	20	12	10	60%	83.33%
Bomba	4	4	4	100%	100%
Peligro	9	7	6	77.78%	85.71%
Total por ciento promedio de reconocimiento.				69.70%	91.30%

En un tercer flujo de audio las palabras fueron pronunciadas de manera pausada y detallada por lo que el porcentaje de exactitud ha vuelto a aumentar hasta 91.30%. Esto demuestra que el mismo varía dependiendo de la forma en que las palabras sean pronunciadas, es decir si son emitidas de manera continua o haciendo énfasis en el fonema a reconocer.

Una vez analizado el comportamiento del sistema para un ambiente controlado le son realizadas pruebas en el ambiente de un sistema de video vigilancia. El mismo es un ambiente no controlado donde la existencia de ruidos alternos paralelos a la voz puede ser bastante elevada y por tanto el reconocimiento de la misma puede tornarse engorroso.

Tabla. 14 Análisis del flujo de audio 4

Palabra pronunciada.	Cantidad de veces pronunciada.	Palabra reconocida	Cantidad de veces reconocida.	Porcentaje de exactitud de reconocimiento.
—	0	bomba	1	0%

La tabla refleja que al captar un flujo de audio en un sistema de video vigilancia la solución a desarrollar reconoce algunos de los ruidos externos como palabras pronunciadas, aún cuando no se ha hablado.

Capítulo 3: Implementación y Prueba

En otra variante del proceso de prueba fueron pronunciadas palabras como: auxilio, peligro y pistola, (ya predefinidas para que sean reconocidas si son pronunciadas) lejos del alcance del micrófono de la cámara y el sistema no las reconoció. Esto demuestra que al añadirle la funcionalidad del reconocimiento de palabras claves a un sistema de video vigilancia se debe tener en cuenta el ambiente en que se desarrollan, por lo que las palabras generalmente no serán pronunciadas de forma pausada y haciendo énfasis en ellas, además de que los ruidos externos simultáneos a la emisión de la voz dificultan en gran medida el reconocimiento de la misma.

3.6.3. Pruebas de Integración

Las pruebas de integración verifican que los componentes de software interaccionan entre sí de forma apropiada después de haber sido integrados en una construcción. La mayoría de los casos de prueba de integración pueden ser derivados de las realizaciones de los CU ya que estas describen cómo interaccionan las clases y los objetos, y por tanto como interaccionan los componentes (Jacobson, y otros, 2000).

En el desarrollo de la presente investigación las pruebas de integración no se realizaron con respecto a los componentes internos del sistema para el reconocimiento de voz sino con respecto a la incorporación del mismo al Sistema de Video Vigilancia Xilema Suria. Al realizar esta operación se procedió a captar el flujo de video que las cámaras visualizan obteniendo de él solo audio. Dicho audio es analizado con el fin de identificar palabras claves en él, no siendo posible esta última acción debido a que los ruidos externos existentes paralelos a la emisión de la voz no posibilitan que esta sea reconocida de manera eficiente.

3.6.4. Pruebas de Aceptación

Al realizar las pruebas de aceptación a un software el sistema debe ser aceptado por el usuario final verificando que el mismo cumpla con el objetivo de la investigación. Por tal motivo son producidos casos de prueba que el sistema deberá pasar satisfactoriamente. Dichos casos de pruebas pueden ser desarrollados durante la fase de diseño e implementación del software. (Alonso Amo, y otros, 2005)

En el desarrollo de la propuesta de solución en curso fueron realizadas las pruebas de aceptación pertinentes referidas al reconocimiento de voz para el sistema de Video Vigilancia Xilema Suria. Los resultados arrojados por las mismas cumplen parcialmente con los resultados esperados pues aunque es

Capítulo 3: Implementación y Prueba

analizado el flujo de audio captado por la cámara se presentan dificultades en el momento del procesamiento para identificar las palabras claves pronunciadas. Aún así la solución que se propone contribuye al desarrollo del centro GEYSED pues la identificación de palabras claves en el audio captado en un ambiente controlado si es posible utilizando la propuesta de la presente investigación.

3.6.5. Resultados de la pruebas de software

Una vez concluido el proceso de pruebas al Video Sensor para el Reconocimiento de Palabras Claves en Audio y analizados los resultados obtenidos se debe tener en cuenta que:

La solución planteada constituye un sistema para la identificación de palabras claves pronunciadas en un ambiente controlado, contando con elevados porcentos de aceptación y exactitud. A pesar de que la misma no puede ser integrada al sistema de Video Vigilancia Xilema Suria si es un aporte para el centro de desarrollo GEYSED al que pertenece dicho proyecto, por lo que la investigación tuvo frutos satisfactorios.

3.7. Conclusiones Parciales

Con la realización de este capítulo se concluye la descripción de la implementación. Le fueron aplicadas pruebas al sistema para garantizar, en su totalidad, el cumplimiento de los requisitos funcionales que se han definido. Se desarrolló el diagrama de componentes que permitió comprender como interaccionan los componentes que forman parte del sistema. Se establecieron estándares de codificación que permitirán el mantenimiento del sistema y una clara visualización del mismo. La aplicación de pruebas determinó que el sistema no presenta inconveniente alguno para el reconocimiento de palabras en un ambiente controlado no siendo así para un ambiente ruidoso. A pesar de ello contribuye al desarrollo del centro GEYSED por la aplicación que puede tener en otros proyectos integrados al mismo.

Conclusiones Generales

Conclusiones Generales

Con la culminación del presente trabajo de diploma se logró dar cumplimiento a las tareas de la investigación propuestas al inicio de la investigación, obteniéndose un video sensor para el reconocimiento de palabras claves en audio, por lo que se arribó a las siguientes conclusiones:

- El análisis de sistemas actuales para el reconocimiento de voz permitió tomar como punto de partida sus principales características para el desarrollo la solución propuesta.
- Para el cumplimiento de los objetivos y en correspondencia con las necesidades expuestas, se requirió hacer un estudio de la situación actual del sistema Video Vigilancia Xilema Suria, lo que garantizó que el audio captado por las cámaras fuera un factor indispensable durante el monitoreo y control del ambiente donde se encuentren instaladas.
- La utilización de la metodología RUP garantizó que en cada iteración del proceso de desarrollo se obtuviera un incremento considerable de la solución propuesta.
- El uso de estilos arquitectónicos y patrones de diseño brindó como resultado una estructura organizacional que respondiera a un diseño sólido y flexible para la codificación del sistema.
- Los resultados arrojados por las pruebas realizadas cumplieron con las expectativas del equipo de desarrollo y aunque no se logró la integración de la solución al sistema de Video Vigilancia Xilema Suria la propuesta de solución constituye un aporte al centro de desarrollo GEYSED.

Recomendaciones

Durante el desarrollo del presente trabajo han surgido ideas que podrían implementarse, de forma que se lograría incrementar las funcionalidades del sistema ampliando así su funcionamiento, para lo cual se recomienda:

- Optimizar el funcionamiento del presente sistema de reconocimiento de voz para su desarrollo en un ambiente ruidoso.
- Enriquecer el diccionario que contiene las palabras claves que son reconocidas en la solución propuesta.
- Implementar un algoritmo que aumente la eficiencia del análisis del diccionario utilizado para el reconocimiento de palabras claves.

Referencias Bibliográficas

Referencias

Accesor Applications and Services, S.A. 2014. Accesor Soluciones en control de accesos y seguridad. *Accesor Soluciones en control de accesos y seguridad*. [En línea] Accesor Applications and Services, S.A, 2 de 2014. [Citado el: 19 de 2 de 2014.] www.accesor.com.

Alonso Amo, F, Martinez Normand, Loic y Segovia Pérez, Francisco Javier. 2005. *Introducción a la ingeniería del software*. s.l. : Delta Publicaciones, 2005. 8496477002.

Bhanu, Bir, y otros. 2011. *Distributed Video Sensor Networks*. s.l. : Springer, 2011.

Bravo, José y Ortega, Manuel. 2000. *Línea de investigación en informática*. España : Universidad de Castilla La Mancha, 2000. 8484270289.

Bustos, Alberto. 2011. Dudas Linguisticas. *Dudas Linguisticas*. [En línea] 5 de 12 de 2011. [Citado el: 19 de 2 de 2014.] www.lengua-e.com.

C++ con clase . 2007. C++ con clase. *C++ con clase*. [En línea] 2007. <http://c.conclase.net/>.

Díaz Verdejo, J. E., y otros. 2012. *Entrenamiento Discriminativo para HMM utilizando redes Neuronales*. Granada : Dept. de Electrónica y tecnología de computadoras, 2012.

Downey, Allen B. 2012. *Pensando en la computación como un científico (Con Java)*. Buenos Aires : Colección Ciencia, Innovación y Sociedad, 2012. 9789876301176.

Eclipse Foundation. 2014. Eclipse. *Eclipse*. [En línea] Eclipse Foundation, 2014. [Citado el: 4 de 3 de 2014.] www.eclipse.org.

Falgueras, Benet Campderrich. 2002. *Ingeniería del software*. s.l. : UOC, 2002. 8484297934.

Fernández, Julian Rodríguez. 2013. *Circuito Cerrado de Televisión y Seguridad Electrónica*. Madrid : Paraninfo, 2013. 9788497326698.

García, Ángel López. 1999. *Lingüística General y Aplicada (3ra Edición)*. Valencia : Universitat de Valencia, 1999. 8437015197.

García, Miguel Barceló. *Inteligencia Artificial*. s.l. : Editorial UOC. 8484298930.

Grijalva, William Castro. 2003. *Análisis de voz usando Codificación Predictiva Lineal*. Lima : Universidad Nacional de Ingeniería, 2003.

Iborra, Álvaro Navarro. 2008. *Una interfaz vocal para la detección de la narración*. Madrid : Facultad Universidad Complutense, 2008.

Mata, Francisco Javier García. 2010. *CCTV usando video IP*. s.l. : Vértice, 2010. 8499313566.

Microsoft Windows. 2011. Microsoft Windows. *Microsoft Windows*. [En línea] Microsoft Windows, 29 de 8 de 2011. [Citado el: 9 de 12 de 2013.] www.deskshare.com.

Referencias Bibliográficas

Paniagua, Enrique y López Ayuso, Belén. 2007. *La gestión Tecnológica del conocimiento.* s.l. : EDITUM, 2007. 8483716615.

Pressman, Roger. 1997. *Ingeniería de Software un enfoque práctico 5ta Edición.* Madrid : Universidad Pontificia de Salamanca. Campus de Madrid, 1997.

Qt Project Hosting SA. 2013. Qt Project Hosting. *Qt Project Hosting.* [En línea] 2013. [Citado el: 9 de 12 de 2013.] <http://qt-project.org>.

Robledo Sacristán, Clodoaldo y Robledo Fernández, David. 2012. *Programación en Android.* s.l. : Ministerio de Educación, 2012. 8436954319.

Rodriguez Oropeza, José Luis. 2006. *Algoritmos y Métodos para el reconocimiento de voz en español mediante sílabas.* Madrid : Centro de Investigación en Computación -IPN, 2006. 14055546.

Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 2000. *El Proceso Unificado de Desarrollo de Software.* Madrid : Pearson Education.S.A, 2000. 8478290362.

S.A, Targetware. 2013. software. *software.* [En línea] Targetware S.A., 5 de 11 de 2013. [Citado el: 9 de 12 de 2013.] www.software.com.ar.

Sánchez, Jesus Vallinot. 2008. *Módulo de Transcripción VoIP-Texto para una plataforma legal de comunicación.* Madrid : Facultad Universidad Complutense, 2008.

Sommerville, Ian. 2005. *Ingeniería del Software.* s.l. : Pearson Educación, 2005. 8478290745.

Bibliografía

Accesor Applications and Services, S.A. 2014. Accesor Soluciones en control de accesos y seguridad. *Accesor Soluciones en control de accesos y seguridad*. [En línea] Accesor Applications and Services, S.A, 2 de 2014. [Citado el: 19 de 2 de 2014.] www.accesor.com.

Alonso Amo, F, Martinez Normand, Loic y Segovia Pérez, Francisco Javier. 2005. *Introducción a la ingeniería del software*. s.l. : Delta Publicaciones, 2005. 8496477002.

Bhanu, Bir, y otros. 2011. *Distributed Video Sensor Networks*. s.l. : Springer, 2011.

Bravo, José y Ortega, Manuel. 2000. *Línea de investigación en informática*. España : Universidad de Castilla La Mancha, 2000. 8484270289.

Bustos, Alberto. 2011. Dudas Linguisticas. *Dudas Linguisticas*. [En línea] 5 de 12 de 2011. [Citado el: 19 de 2 de 2014.] www.lengua-e.com.

C++ con clase . 2007. C++ con clase. *C++ con clase*. [En línea] 2007. <http://c.conclase.net/>.

Center, Virtual Reality Applications. 2011. *Open Source Speech Interaction with the Voice*. Ames : Iowa State University, 2011.

Díaz Verdejo, J. E., y otros. 2012. *Entrenamiento Discriminativo para HMM utilizando redes Neuronales*. Granada : Dept. de Electrónica y tecnología de computadoras, 2012.

Downey, Allen B. 2012. *Pensando en la computación como un científico (Con Java)*. Buenos Aires : Colección Ciencia, Innovación y Sociedad, 2012. 9789876301176.

—. 2012. *Pensando en la Computación como un Científico (Con Java)*. Buenos Aires : Colección Ciencia, Innovación y Desarrollo, 2012. 9789876301176.

Eclipse Foundation. 2014. Eclipse. *Eclipse*. [En línea] Eclipse Foundation, 2014. [Citado el: 4 de 3 de 2014.] www.eclipse.org.

Falgueras, Benet Campderrich. 2002. *Ingeniería del software*. s.l. : UOC, 2002. 8484297934.

Fernández, Julian Rodríguez. 2013. *Circuito Cerrado de Televisión y Seguridad Electrónica*. Madrid : Paraninfo, 2013. 9788497326698.

García, Ángel López. 1999. *Lingüística General y Aplicada (3ra Edición)*. Valencia : Universitat de Valencia, 1999. 8437015197.

García, Miguel Barceló. *Inteligencia Artificial*. s.l. : Editorial UOC. 8484298930.

Grijalva, William Castro. 2003. *Análisis de voz usando Codificación Predictiva Lineal*. Lima : Universidad Nacional de Ingeniería, 2003.

Iborra, Álvaro Navarro. 2008. *Una interfaz vocal para la detección de la narración*. Madrid : Facultad Universidad Complutense, 2008.

Bibliografía

- Mata, Francisco Javier García. 2010.** *CCTV usando video IP*. s.l. : Vértice, 2010. 8499313566.
- Microsoft Windows. 2011.** Microsoft Windows. *Microsoft Windows*. [En línea] Microsoft Windows, 29 de 8 de 2011. [Citado el: 9 de 12 de 2013.] www.deskshare.com.
- Paniagua, Enrique y López Ayuso, Belén. 2007.** *La gestión Tecnológica del conocimiento*. s.l. : EDITUM, 2007. 8483716615.
- Pressman, Roger. 1997.** *Ingeniería de Software un enfoque práctico 5ta Edición*. Madrid : Universidad Pontificia de Salamanca. Campus de Madrid, 1997.
- Qt Project Hosting SA. 2013.** Qt Project Hosting. *Qt Project Hosting*. [En línea] 2013. [Citado el: 9 de 12 de 2013.] <http://qt-project.org>.
- Robledo Sacristán, Clodoaldo y Robledo Fernández, David. 2012.** *Programación en Android*. s.l. : Ministerio de Educación, 2012. 8436954319.
- Rodriguez Oropeza, José Luis. 2006.** *Algoritmos y Métodos para el reconocimiento de voz en español mediante sílabas*. Madrid : Centro de Investigación en Computación -IPN, 2006. 14055546.
- Rodriguez, José Luis Oropeza. 2006.** *Algoritmos y Metodos para el reconocimiento de Voz en Español Mediante Sílabas*. España : Centro de Investigación en Computación-IPN, 2006. 1405-5546.
- Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 2000.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Education.S.A, 2000. 8478290362.
- S.A, Targetware. 2013.** software. *software*. [En línea] Targetware S.A., 5 de 11 de 2013. [Citado el: 9 de 12 de 2013.] www.software.com.ar.
- Sánchez, Jesus Vallinot. 2008.** *Módulo de Transcripción VoIP-Texto para una plataforma legal de comunicación*. Madrid : Facultad Universidad Complutense, 2008.
- Sommerville, Ian. 2005.** *Ingeniería del Software*. s.l. : Pearson Educación, 2005. 8478290745.

Anexos

Anexo 1: Entrevista

Durante la entrevista como técnica para obtener información asociada al sistema Video Vigilancia Xilema Suria y a pesar de ser de tipo no estructurada, se realizaron una serie de preguntas. Esta acción tuvo como propósito obtener un conocimiento detallado de la situación del proyecto, en función de llevar a cabo el reconocimiento de voz a través de un video sensor.

Fecha: 14 de febrero de 2014.

Nombre de las personas entrevistadas:

- Reynier Pupo Gómez.
- Jean Michel Suárez Pérez.
- Reinier Pupo Ruiz.

1. ¿Cómo funciona el sistema de Video Vigilancia Xilema Suria con respecto a los módulos que lo componen?
2. ¿Cómo procesa el Módulo de Análisis el flujo de video captado por cámaras IP?
3. ¿Cuáles son los tipos de video sensores desarrollados por el sistema Video Vigilancia Xilema Suria?
4. ¿De los protocolos existentes para la transmisión del flujo de video captado por las cámaras, cuál es el más seguro?
5. ¿Cuáles son las herramientas existentes para el reconocimiento de palabras claves en audio?

Anexos

Anexo 2: Descripción de los Algoritmos fundamentales implementados.

Tabla. 15 Algoritmos fundamentales implementados.

Nombre de la Clase	Nombre del Método	Descripción
CC_Reconocedor	Capturar()	Esta funcionalidad es la encargada de capturar el flujo video que está grabando la cámara en tiempo real y extraer el audio de dicho flujo.
CC_Reconocedor	Reconocer()	Esta funcionalidad es la encargada de que una vez que el audio es almacenado de manera temporal en una dirección específica, captarlo de dicha dirección y ejecutar el proceso que sea capaz de transcribirlo a texto.
CC_Reconocedor	TextoReconocido()	Es quien captura el flujo de texto con las palabras significativas mencionadas en el audio y las envía al módulo de Análisis.
CC_Transcriber	main(String[]: args)	Este método es el encargado de transcribir el flujo de audio transmitido por las cámaras de video vigilancia.