



UNIVERSIDAD
DE LOS ANDES
VENEZUELA

PROYECTO DE GRADO

Presentado ante la ilustre UNIVERSIDAD DE LOS ANDES como requisito parcial para
obtener el Título de INGENIERO DE SISTEMAS

DESARROLLO DE UN GESTOR DE INFORMACIÓN PARA PERSONAS CON DISCAPACIDADES DE VISIÓN CON EL FIN DE FACILITAR LA ACCESIBILIDAD Y GESTIÓN DE LA INFORMACIÓN EN INTERNET

Por

Br. Carlos Arturo Rodríguez Pacheco

Tutor: Prof. Rafael Rivas Estrada

Asesor: MSc. Ana Corrales Paredes

Marzo 2010

©2010 Universidad de Los Andes, Mérida, Venezuela

Desarrollo de un gestor de información para personas con discapacidades de visión con el fin de facilitar la accesibilidad y gestión de la información en Internet

Br. Carlos Arturo Rodríguez Pacheco

Proyecto de Grado — Sistemas Computacionales, 77 páginas
Escuela de Ingeniería de Sistemas, Universidad de Los Andes, 2010

Resumen: El presente trabajo presenta el diseño y desarrollo de un componente de software en pro de las personas discapacitadas, en particular personas con problemas de visión o invidentes. Se propone un gestor de información capaz de procesar información inteligentemente y suministrarla a un usuario final, con el fin de facilitar el acceso al contenido de los sitios *web*, que utilizan la tecnología *Really Simple Syndication* (RSS). Este proyecto de grado está enmarcado dentro del proyecto *Maggie Personal Robots* del laboratorio *Robotics Lab* de la Universidad Carlos III de Madrid, España. Uno de los objetivos de este trabajo es incorporar al robot personal Maggie la habilidad de adquirir la información actualizada de las *web's* con RSS y suministrarla al usuario cuando este lo solicite.

Palabras clave: Gestor de Información, RSS, Sintetizador de voz, Reconocedor de voz.

www.bdigital.lula.ve *A Dios y a mi familia.*

Índice general

Índice de Tablas	VII
Índice de Figuras	IX
Agradecimientos	XI
1. Aspectos Preliminares	1
1.1. Antecedentes	1
1.2. Justificación	6
1.3. Planteamiento del problema	6
1.4. Objetivos	7
1.4.1. Objetivo general	7
1.4.2. Objetivos específicos	7
1.5. Metodología	8
1.5.1. Primera fase	8
1.5.2. Segunda fase	9
1.6. Estructura del documento	9
2. Nociones Teóricas	10
2.1. Gestor de información	10
2.2. Robot personal <i>Maggie</i>	11
2.2.1. Diseño del robot <i>Maggie</i>	12
2.2.2. Arquitectura de <i>software</i> del robot <i>Maggie</i>	13
2.3. Método <i>Watch Component</i>	14
2.4. XML	17

2.4.1.	Ventajas de XML	18
2.5.	RSS	18
2.5.1.	Utilidad de los RSS	19
2.6.	Síntesis y reconocimiento de voz	20
2.6.1.	Sintetizador de voz	21
2.6.2.	Reconocedor de voz	22
2.6.3.	Componentes de <i>software</i> para la síntesis y reconocimiento de voz	22
3.	Gestor de información, especificación, requerimientos, diseño y aprovisionamiento	24
3.1.	Especificación del producto de <i>software</i>	25
3.2.	Análisis y especificación de requerimientos	27
3.2.1.	Casos de uso	29
3.2.2.	Diagramas de actividades y diagramas de secuencias	34
3.3.	Diagrama de clases y entidad relación	37
3.4.	Diseño del producto de <i>software</i>	40
3.4.1.	Diseño de la arquitectura del sistema	40
3.4.2.	Diseño de las interfaces auditiva y gráfica del sistema	43
3.4.3.	Diseño de pantallas	44
3.5.	Aprovisionamiento de los componentes de <i>software</i>	46
3.5.1.	Adquisición de los componentes de <i>software</i>	46
4.	Gestor de información, pruebas, certificación y liberación	49
4.1.	Pruebas del componente de <i>software</i>	50
4.1.1.	Pruebas caja negra	50
4.1.2.	Pruebas funcionales	51
4.1.3.	Prueba de instalación	55
4.1.4.	Prueba de recuperación	55
4.2.	Certificación del componente de <i>software</i>	56
4.3.	Liberación del componente de <i>software</i>	57

5. Conclusiones y recomendaciones	58
5.1. Conclusiones	58
5.2. Recomendaciones	59
Bibliografía	60
A. Ajuste del método <i>Watch Component</i>	63
B. Tablas de especificación de requisitos	64
C. Diagramas de secuencia	70
D. Diagramas de clase implementado y tablas de especificación de las clases	72
E. Capturas de pantallas de la interfaz grafica de configuración	75

www.bdigital.ula.ve

Índice de tablas

3.1. Tabla de requisitos del gestor de información.	28
3.2. Especificación, caso de uso transmitir noticias.	31
3.3. Especificación, caso de uso configurar las fuentes RSS.	33
3.4. Comparación de los componentes sintetizadores y reconocedores de voz.	47
3.5. Comparación de los componentes de <i>software</i>	47
4.1. Pruebas caja negra para las funciones y métodos del gestor de información	51
4.2. Verificación de requisitos cumplidos	54
4.3. Pruebas de integración de los componentes	56
B.1. Especificación del requisito 1	64
B.2. Especificación del requisito 2	65
B.3. Especificación del requisito 3	65
B.4. Especificación del requisito 4	65
B.5. Especificación del requisito 5	66
B.6. Especificación del requisito 6	66
B.7. Especificación del requisito 7	66
B.8. Especificación del requisito 8	67
B.9. Especificación del requisito 9	67
B.10. Especificación del requisito 10	67
B.11. Especificación del requisito 11	68
B.12. Especificación del requisito 12	68
B.13. Especificación del requisito 13	68
B.14. Especificación del requisito 14	69
B.15. Especificación del requisito 15	69

D.1. Especificación de la clase configuración.	72
D.2. Especificación de la clase gestor información.	74

www.bdigital.ula.ve

Índice de figuras

1.1. Gráfico de la evolución de la tecnología (Baekdal 2009)	2
1.2. Robot <i>Maggie</i> , desarrollado por la UC3M	5
1.3. <i>Nabaztag</i> , creado por <i>Violet</i> (Violet 2007)	6
2.1. El Robot Personal <i>Maggie</i> , dando las noticias. IX Feria Madrid es Ciencia 2008 (Corrales et al. 2008)	11
2.2. El Robot Personal <i>Maggie</i>	12
2.3. Arquitectura del Robot Personal <i>Maggie</i>	13
2.4. Método <i>Watch Component</i>	15
2.5. Proceso espiral para el análisis de los requisitos.	16
2.6. Código XML	17
2.7. Código fuente del estándar de los RSS	19
2.8. Icono de identificación de los RSS	19
2.9. Componentes de <i>software</i> para la síntesis y reconocimiento de voz.	23
3.1. Método <i>Watch Component</i> , especificación, requerimientos, diseño y aprovisionamiento.	24
3.2. Diagrama de funcionamiento del gestor de información	26
3.3. Actores del sistema.	27
3.4. Diagrama del uso general del gestor de información.	29
3.5. Diagrama caso de uso buscar noticias.	30
3.6. Diagrama caso de uso transmitir noticias.	31
3.7. Diagrama caso de uso configurar las fuentes RSS.	32
3.8. Diagrama de flujos alternativos del caso de uso configurar las fuentes RSS.	32
3.9. Diagrama de actividad búsqueda de información.	35

3.10. Diagrama de secuencia búsqueda exitosa.	36
3.11. Diagrama conceptual de clases.	38
3.12. Diagrama entidad relación.	39
3.13. Diagrama de bloques, arquitectura general del sistema.	41
3.14. Diagrama de componentes, arquitectura del sistema.	42
3.15. Captura de pantalla, página principal.	44
3.16. Captura de pantalla, configuración del sistema.	45
4.1. Método Watch component	49
4.2. Captura de pantalla, búsqueda.	52
4.3. Captura de pantalla, base de datos.	53
4.4. Captura de pantalla, infoXML.xml	53
4.5. Captura de pantalla, documento noticia.txt	53
A.1. Ajuste del método <i>Watch Component</i>	63
C.1. Diagrama de secuencia noticia anterior.	70
C.2. Diagrama de secuencia noticia siguiente.	71
C.3. Diagrama de secuencia detener el gestor de información.	71
D.1. Diagrama de clases.	73
E.1. Captura de pantalla, insertar.	75
E.2. Captura de pantalla, modificar.	76
E.3. Captura de pantalla, eliminar.	76
E.4. Captura de pantalla, listar.	77

www.bdigital.ula.ve

Agradecimientos

A Dios, a la virgen y al sagrado corazón de Jesús por guiarme y llenarme de bendiciones durante este camino.

A mis padres por su amor, apoyo incondicional, ser un gran modelo y ejemplo de vida a seguir, también por darme todas las herramientas necesarias para seguir adelante. ¡Los quiero mucho!

A mis hermanos por apoyarme y siempre estar a mi lado. ¡Los quiero mucho!

A mis abuelos, tíos y padrinos gracias por sus bendiciones.

A Angélica por su cariño, comprensión, apoyo y estar siempre presente. ¡Te quiero mucho!

Al Prof. Rafael Rivas por su dedicación y apoyo durante mi trayecto estudiantil.

A Ana Corrales a pesar de la distancia siempre presente apoyándome y guiándome durante el desarrollo de este proyecto de tesis.

A mis amigos y compañeros con quienes he compartidos grandes momentos, se les quiere.

Capítulo 1

Aspectos Preliminares

1.1. Antecedentes

La evolución tecnológica a partir de la década de los cuarenta impulsó un extraordinario avance gracias al diseño y progresividad de los computadores. El computador se ha definido como una máquina capaz de realizar y controlar a gran velocidad cálculos y procesos complicados que requieren una toma rápida de decisiones. Una de sus principales funciones es tratar la información que se le suministra y proveer los resultados requeridos. Por otro lado, los continuos avances en los campos de la robótica y la inteligencia artificial dan pie a las fantasías más futuristas que la mente humana pueda imaginar.

Hoy por hoy, el computador es incapaz de hacer algo para lo cual no ha sido programado, sin embargo su extraordinaria versatilidad en todos los campos de las actividades humanas han hecho de esta máquina una herramienta primordial en gran parte del público.

Un mundo en constante cambio, existe un gran momento de desarrollo tecnológico, donde se cuenta con diversas fuentes de información que están al alcance de todos.

La necesidad del hombre para mantenerse informado puede ser saciada de múltiples maneras dejando a un lado los medios de comunicación más frecuentes como periódicos, observar y escuchar los noticieros de radio y televisión, donde la información cumple un patrón de secuencia preestablecido por el medio y el usuario no tiene la libertad de

escoger información sobre temas de su interés. Por ello existen nuevas formas de fuentes de información alternativas, que están sustituyendo la manera clásica de informar, donde el cliente decide qué tipo de información quiere y cuándo la desea consultar.

Internet se ha convertido en la actualidad en plataforma tecnológica de punta en el desarrollo de fuente de información, ya que le permite al usuario mantenerse informado a cualquier hora y en cualquier lugar. Si se observa la Internet como un nuevo medio de comunicación, éste posee una gran diversidad de formas para transmitir información (sitios *web* y *blogs* que abarcan temas de cualquier índole). Las redes sociales como *Myspace*, *Twitter* y *Facebook* en Internet le permiten al usuario ser el reportero de la información y compartirla con otros usuarios.

En la figura 1.1 se puede observar como ha sido el avance de la tecnología con el pasar del tiempo.

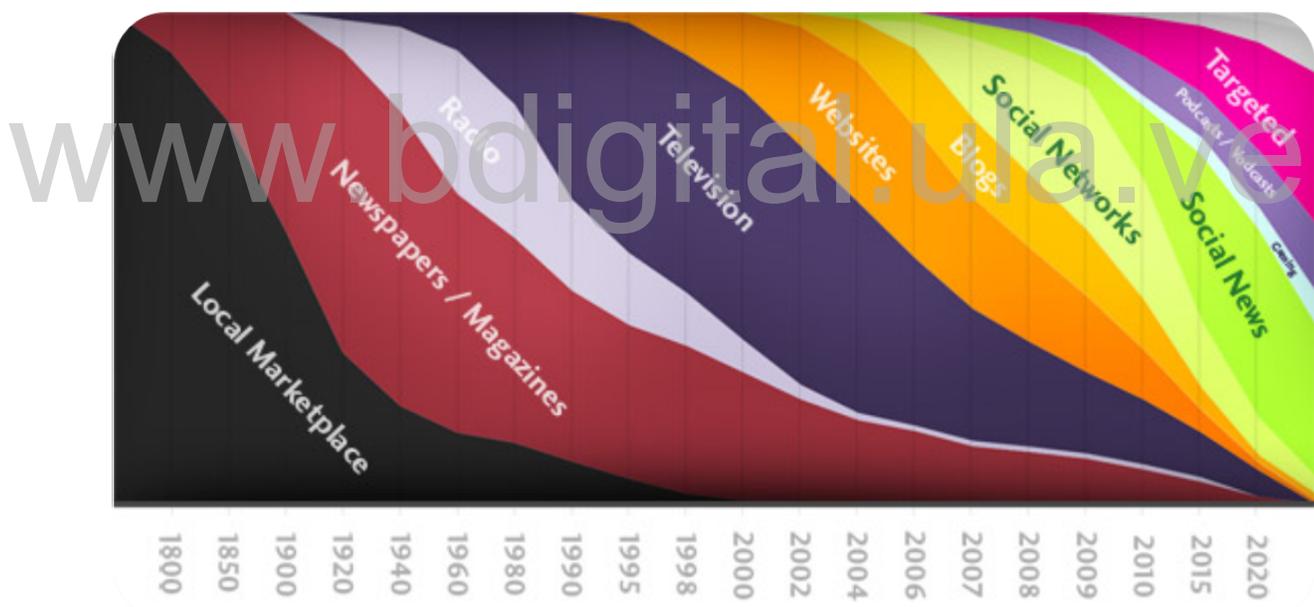


Figura 1.1: Gráfico de la evolución de la tecnología (Baekdal 2009)

El creciente uso de las tecnologías de información, satisface en parte a las necesidades del hombre en mantenerse informado. Esto se logra a través de una gran diversidad de medios de comunicación, pero en su mayoría, estos medios no están en la capacidad de informar a personas con discapacidades visuales.

Existen millones de personas con discapacidades en el mundo, como lo son las

personas invidentes, quienes en muchos de los casos, sólo interactúan a través de medios como la radio y audio de la televisión para informarse.

Dichas personas tienen una gran limitación de informarse sobre temas de su interés, y más aún, poder utilizar la *web* como servicio de información. Los computadores ayudan a los ciegos a ver, en realidad esto no significa que recuperen su visión, pero si gracias al uso de dispositivos informatizados pueden ser más activos y hacer cosas que antes no les estaban permitidas.

Uno de los primeros avances en el desarrollo de tecnología para personas con discapacidades de visión fue el *Optacon* (Nauta 1995), este dispositivo es capaz de leer las letras de una página y traducir en impulsos, que puedan hacerse sentir en la piel de las personas invidentes.

También está la máquina desarrollada por *Kurzweil Computer Products* de Cambridge que lee en voz alta un libro situado debajo de sus dispositivos de exploración. Las letras de las páginas son convertidas en señales digitales que analiza un pequeño computador y un sintetizador de palabra electrónico transforma en voz.

Estos dispositivos de investigación marcan pauta en la historia, dando inicio a nuevos desarrollos de tecnologías para personas con diversos tipos de discapacidades (visión, audición, motores).

En la actualidad se emplean láseres, tecnología informática en el desarrollo de lentes, lápices especiales que traducen los sonidos de los objetos a modelos que permiten al ciego determinar si delante de él hay algún objeto, impresoras en braille, mejoras en reconocedores de voz de manera que sean más versátiles al momento de hacer su trabajo.

Los paquetes de *software* desarrollados para personas con discapacidades, en gran parte han sido desarrollados bajo licencias privativas, que sólo brindan versiones de prueba gratuitas con funcionalidades limitadas, el costo de sus licencias es bastante elevado siendo en algunos casos inaccesibles para las personas de bajos recursos económicos.

Hasta hace pocos años atrás los sitios *web* y el *software* utilizado en Internet presentaban grandes barreras de accesibilidad, lo que dificultaba o imposibilitaba la utilización de la *web* para muchas personas con discapacidad.

Las nuevas tecnologías, sin embargo, ofrecen cada día mayores posibilidades a las

personas discapacitadas para navegar, desde las más sencillas a las más complejas. Desde poder aumentar el tipo de letra y dotar a la pantalla de un mejor contraste, para que las personas con visión insuficiente puedan leer con facilidad, hasta poder convertir los textos en voz a través de programas de sintetización cada vez más perfeccionados.

Cuanto mayor sea la accesibilidad disponible en *software* y sitios en Internet, más personas con discapacidad podrán utilizar la *web* y contribuir aportando nuevas ideas para futuros desarrollos en este ámbito así como también mejorar los ya existentes de una forma más eficiente en relación a su uso.

La *web* es un recurso muy importante para diferentes aspectos de la vida: educación, empleo, gobierno, comercio, sanidad, entretenimiento y muchos otros. Es muy importante que la *web* sea accesible para así proporcionar igualdad de oportunidades a las personas con discapacidad (W3C 2007a).

Existen organizaciones sin fines de lucro a nivel mundial que han colaborado con este tipo de desarrollos y están dispuestas a seguirlo haciendo. En Venezuela se tiene la Fundación Mevorah Florentin y Venezuela sin límites entre otras organizaciones en pro de la integración a la sociedad de las personas con discapacidades de visión. El Centro Nacional de Tecnologías de Información (CNTI) a través de la Red de Infocentros son medios fundamentales de tecnología, especializados para responder a la necesidad de información de personas no videntes o de baja visión.

Por otra parte, la Fundación ONCE es una Corporación sin fines de lucro con la misión de mejorar la calidad de vida de las personas ciegas y con discapacidad visual de toda España (ONCE 2008). Esta fundación ha puesto en marcha la Fundación ONCE para América Latina (FOAL) que ha dado inicio del proyecto global que está permitiendo consolidar el movimiento asociativo de los ciegos latinoamericanos. Asimismo, FOAL contribuye a la creación de empleo y de mejoras educativas para los ciegos de los países de latinoamerica (FOAL 2006).

Tomando como referencia el artículo publicado por Mediavilla (2008) periodista, presidente y editor del diario *Madridiario.es* en el blog *Com.permiso* indica que Internet es una ciudad virtual donde hay barreras arquitectónicas que dificultan a los discapacitados poder salir, entrar o simplemente pasear por sus calles, sus autopistas, o sus rincones favoritos.

Previas investigaciones a este trabajo describen los desarrollos diseñados y contruidos sobre el robot *Maggie*. Éste es un robot personal desarrollado por el Laboratorio de *Robotics lab* de la Universidad Carlos III de Madrid (UC3M) (ver figura 1.2)).



Figura 1.2: Robot *Maggie*, desarrollado por la UC3M

Se trata de una plataforma de investigación para estudiar la interacción humano-robot, la inteligencia y autonomía de los robots (ver capítulo 2). Es un robot creado para socializar con gente, para informar, entretener y ayudar (Robotics-Lab 2006).

En el mercado han aparecido productos con funcionalidades similares, como lo es el *Nabaztag* (Violet 2007). Es una interfaz *WiFi* con forma de conejo amigable (Figura 1.3), que presenta informaciones similares a los RSS pero con protocolos privados y servidores de publicaciones propios, tales que para acceder a todas las funcionalidades el usuario debe suscribirse y pagar por los servicios (Corrales et al. 2008).



Figura 1.3: *Nabaztag*, creado por *Violet* (Violet 2007)

1.2. Justificación

En el mundo existe un porcentaje significativo de la población con muchos tipos de discapacidades, en particular personas con dificultades parciales o totales de visión, quienes se les presentan barreras tecnológicas para tener acceso a la información en Internet.

Este proyecto busca garantizar y mejorar los mecanismos de acceso a la información que circula en Internet a usuarios con discapacidades de visión, tercera edad e inclusive usuarios en general.

1.3. Planteamiento del problema

Teniendo en cuenta el incremento y la importancia de Internet en los últimos años, se plantea desarrollar un gestor de información para usuarios con requerimientos especiales, principalmente orientado a personas con discapacidades de visión, que le imposibilitan el uso de dispositivos periféricos como teclados o monitores.

Este gestor tiene como principal característica, brindar gran ayuda a personas invidentes. Gracias a éste los usuarios podrán obtener información de búsqueda en Internet a través de comandos de voz, así como interactuar de una manera más directa

con el computador.

Todo esto con la finalidad de otorgar acceso a la red a personas que por sus discapacidades se les hace casi imposible el acceso a la información que allí se encuentra.

Desde el punto de vista de *hardware* el sistema a desarrollar puede ser implementado en un computador que posea periféricos multimedia (microfono y altavoces) y por supuesto ser implementado en robots personales como lo es *Maggie*.

1.4. Objetivos

A continuación se presenta el objetivo general y los objetivos específicos del proyecto, estos elementos identifican cual es la finalidad hacia la cual deben dirigirse los recursos y esfuerzos para dar cumplimiento a los propósitos planteados.

1.4.1. Objetivo general

Desarrollar un gestor de información para personas con discapacidades de visión con el fin de facilitar la accesibilidad y gestión de la información en Internet.

1.4.2. Objetivos específicos

- Revisar el estado del arte de los diferentes componentes de *software* existentes para personas con discapacidades de visión.
- Estudiar los diferentes lenguajes presentes en el desarrollo del *software*: *XML, C++, Python, PHP, MySQL*.
- Buscar cuál es la metodología de desarrollo de *software* que mejor se adapta a la solución del problema.
- Buscar alternativas de *software* de reconocimiento y síntesis de voz.
- Estudiar en qué consiste el funcionamiento de los RSS como fuente de información.

- Estudiar la integración de un sistema manejador de base de datos y los componentes de RSS.
- Integrar el *software* de reconocimiento y sintetizador de voz con el sistema de búsqueda en la base de datos.
- Integrar el gestor de información con la interfaz gráfica.

1.5. Metodología

La metodología empleada para el desarrollo de este trabajo se divide en dos grandes fases, la primera fase todo lo referente al estado del arte del *software* correspondiente, utilización de los diferentes lenguajes y el estudio de la integración del *software*. Por otra parte, la segunda fase abarca al desarrollo de la aplicación utilizando el método *Watch* de desarrollo de *software* y componentes. A continuación se desglosa cada fase:

1.5.1. Primera fase

- Revisión del estado del arte de *software*, aplicaciones y componentes existentes para personas con discapacidades de visión.
- Estudio en profundidad de los lenguajes, el servidor de base de datos, para el desarrollo del proyecto, *C++*, *Python*, *XML*, *PHP*, *MySQL*.
- Estudio de las fuentes RSS en qué consisten y como es su funcionamiento como una fuente de información.
- Estudio e integración de la base de datos desarrollada en *MySQL* con los componentes de fuentes de información RSS.
- Búsqueda y verificación de alternativas de *software* de reconocimiento y sintetizador de voz.

1.5.2. Segunda fase

Durante este ciclo del método *Watch Component* se desglosa y desarrolla de la siguiente forma(En el capítulo 2 se describe con mayor profundidad el desarrollo del método *Watch Component*):

- Especificación del componente.
- Aprovisionamiento del componente.
- Pruebas del componente.
- Certificación del componente.
- Liberación del componente.

1.6. Estructura del documento

El presente documento ha sido dividido en cinco capítulos. En el primer capítulo se presentan los antecedentes, el planteamiento del problema, las razones que lo justifican, los objetivos planteados, la metodología a seguir y la estructura del documento.

En el segundo capítulo se presentan todas las definiciones teóricas necesarias que sustentan este trabajo.

En el tercer capítulo se presenta la especificación y aprovisionamiento del componente. Fases de la metodología *Watch Component* para la construcción del gestor de información. También se presentan los diagramas UML que son los planos del sistema (casos de uso, diagramas de componentes y arquitectura, diagrama de clases, diagrama de actividades y diagramas de secuencia) que son pieza fundamental al entendimiento e implantación del *software*, Así como también el diseño del producto de software y aprovisionamiento de componentes.

En el cuarto capítulo se presentan las pruebas que garantizan funcionamiento y comportamiento del sistema ante cualquier fallo, verificación y liberación del componente de *software*.

Finalmente en el quinto capítulo se presentan las conclusiones obtenidas del presente trabajo y algunas recomendaciones útiles para trabajos futuros.

Capítulo 2

Nociones Teóricas

Este capítulo presenta todas las definiciones teóricas necesarias que sustentan y facilitan el entendimiento del componente de *software* desarrollado en este proyecto

2.1. Gestor de información

Basado en páginas *web* con formatos de datos RSS, el gestor de información es un sistema capaz de buscar, procesar, actualizar y suministrar la información de manera eficiente al usuario final.

El gestor de información es una herramienta que facilita al usuario el servicio de información de sitios *webs* con contenidos RSS. Este componente de *software* le permite al usuario obtener de manera veloz información actualizada de sus sitios *webs* favoritos, sin necesidad de acceder a ellos independientemente.

A través de su interfaz auditiva el gestor brinda gran ayuda en la interacción humano-computador (IHC) para usuarios con discapacidades, en particular personas con problemas de visión. Para ello el gestor de información consolida múltiples fuentes de datos (relacionadas entre sí por una categoría determinada) y permite manejarlas de forma unificadas.

2.2. Robot personal *Maggie*

La interacción humano-robot juega un papel muy importante en la incorporación de los robots en la vida diaria. Los robots deben ser capaces entonces de realizar o colaborar en tareas cotidianas (Breazeal 2004). Los robots hasta ahora han sido utilizados en entornos industriales, estructurados y con bajo nivel de autonomía. La interacción humano-robot lleva a tareas más cercanas a usuarios sin conocimientos técnicos, siendo su aplicación en actividades como: asistencia en el hogar, asistencia a personas discapacitadas, robots que sirvan de compañía, asistencia personal, entre otros (Corrales et al. 2008).

Las personas pueden interactuar con *Maggie* hablando, usando gestos o tocándolo. A través de su reconocedor de voz, los usuarios pueden realizar búsquedas y obtener respuestas de forma oral, logrando así estar más informados, e interactuando de una manera más real (Robotics-Lab 2006). En la figura 2.1 puede ver a *Maggie* dando las noticias en la IX Feria Madrid es Ciencia 2008.



Figura 2.1: El Robot Personal *Maggie*, dando las noticias. IX Feria Madrid es Ciencia 2008 (Corrales et al. 2008)

2.2.1. Diseño del robot *Maggie*

El robot *Maggie* tiene un diseño basado en las siguientes consideraciones (Figura 2.2):

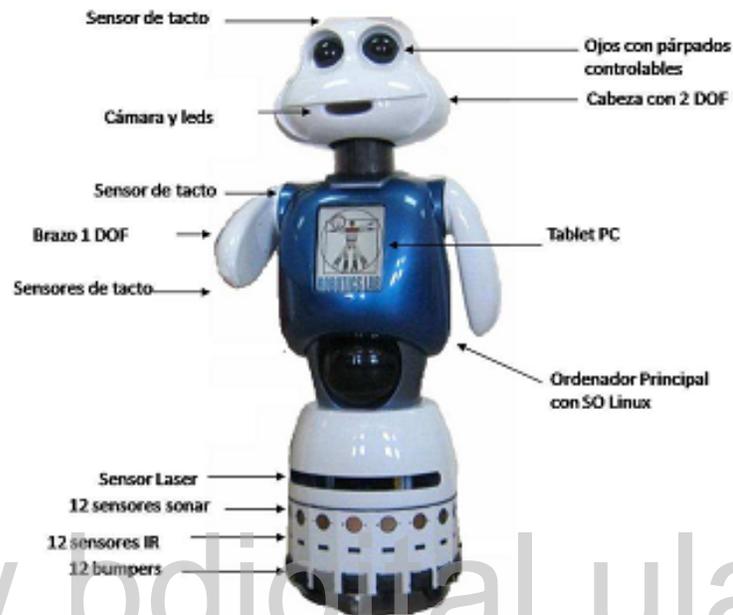


Figura 2.2: El Robot Personal *Maggie*

Atractivo: La apariencia física es importante al interactuar con el robot, *Maggie* posee una apariencia amigable, como un juguete para niños, con 1.35 m de altura y colores agradables a la vista.

Expresividad: *Maggie* no posee expresiones faciales pero puede mover los párpados y tiene una serie de leds azules en la boca que están sincronizados con la voz del robot.

Multimodalidad: Posee diferentes mecanismos de interacción. Tiene varios sensores de tacto distribuidos en diferentes partes del cuerpo, una cámara para procesamiento de imágenes. Para interacción por voz posee altavoces y mediante un micrófono obtiene la información vocal del usuario (Gorostiza et al. 2006).

Movilidad: El movimiento del robot se consigue a través de una base con ruedas y guiado diferencial.

El robot adicionalmente posee 2 lectores de identificación por radiofrecuencia *RFID*

(por sus siglas en inglés *Radio Frequency IDentification*) y un láser *Sick LMS 200*. La cabeza del robot posee una forma antropomórfica con 2 grados de libertad, además tiene 2 brazos de 1 grado de libertad.

Dentro de *Maggie* hay 2 computadores interconectados entre sí a través de una red *Ethernet* y con computadores externos a través de *WiFi*. Uno de los computadores es un *tablet PC* para tareas de interacción y el otro un computador con sistema operativo Linux, que maneja el *software* de los diferentes elementos físicos del robot, y procesa el control y habilidades del robot (Corrales et al. 2008).

2.2.2. Arquitectura de *software* del robot *Maggie*

La arquitectura de software del robot *Maggie* está basada en la arquitectura Automático-Deliberativa AD (Barber 2000), ésta arquitectura consta de dos niveles: deliberativo y automático (ver figura 2.3). El nivel deliberativo está asociado con procesos reflexivos, es decir procesos que requieren de capacidad de razonamiento. El nivel automático se refiere a los procesos automáticos del sistema y está compuesto por los módulos que interactúan con los sensores y actuadores. Ambos niveles están compuestos por habilidades (Corrales et al. 2008).

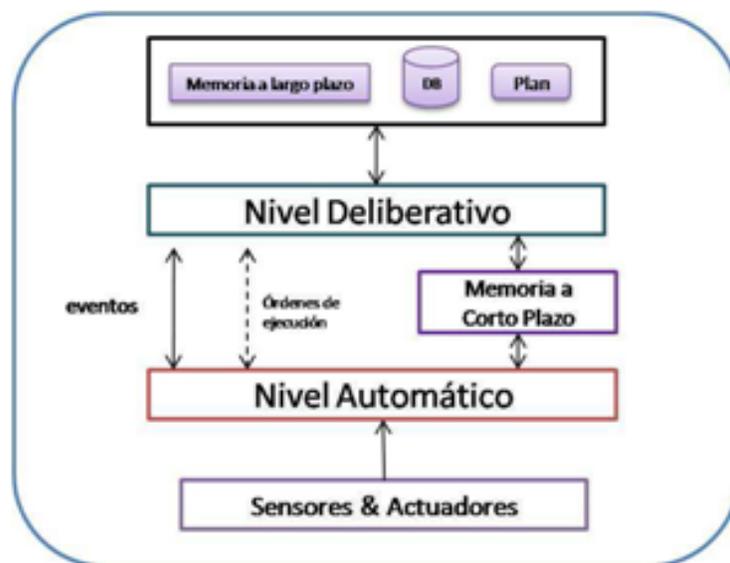


Figura 2.3: Arquitectura del Robot Personal *Maggie*

Una habilidad es el elemento básico de la arquitectura AD. Consiste en la capacidad del robot de realizar un razonamiento, de procesar información o de llevar a cabo una acción. Las habilidades pueden ser activadas por otras habilidades o por un secuenciador de órdenes, devolviendo datos o eventos al elemento que la haya activado o a habilidades que estén interesadas en la información suministrada.

Según R. Rivas (2007), una instancia de un objeto habilidad se caracteriza por:

- Tener 3 estados: lista, en ejecución y bloqueada.
- Tener 3 modos de funcionamiento: continuo, periódico y por eventos.
- Cada habilidad es un proceso, la comunicación entre procesos se hace por medio de eventos o usando la memoria compartida.
- Una habilidad representa una o más tareas o la combinación de diferentes habilidades.
- Una habilidad debe suscribirse a un evento y definir su comportamiento en el momento que el evento ocurre.

La comunicación entre el nivel Automático y el nivel Deliberativo es bidireccional, ambos niveles se comunican a través de la memoria compartida a corto plazo. La memoria a corto plazo guarda la información más importante suministrada por los sensores o por habilidades (Corrales et al. 2008).

2.3. Método *Watch Component*

El desarrollo de este proyecto se hace siguiendo cada una de las fases del método *Watch Component*, una breve explicación de este método para el desarrollo de *software* se presenta a continuación.

Este es un método de desarrollo *software*, el cual describe un conjunto de actividades necesarias para producir componentes (Figura 2.4) (Montilva 2006).

Es importante resaltar que el gestor de información será una nueva habilidad (componente de *software*) para el robot *Maggie* y a su vez una aplicación que puede



Figura 2.4: Método *Watch Component*

ser utilizado independientemente del robot en un computador. Debido a esta particular característica del *software* que se pretende desarrollar, el método de desarrollo ha sido ajustado a las necesidades del gestor de información (ver apendice A figura A.1).

A continuación se explica brevemente el ciclo de reloj desde la especificación del componente, hasta la puesta en marcha del gestor de información.

Especificación del componente: durante esta fase se explica brevemente en que consiste el gestor, requisitos del sistema, alcance, diseño a través de los diferentes diagramas UML. Durante la captación de los requisitos se utilizó un proceso iterativo en forma de espiral descrito en la figura 2.5, el cual fue de gran ayuda, para la especificación de los requisitos necesarios para cumplir los objetivos trazados en este proyecto.

Los planos para la construcción del sistema vienen dados por el diseño del *software*, esto abarca una serie de tareas de codificación y realización de pruebas que conducen al software operativo que está listo para ser entregado al usuario final.

Una vez construido el producto de software siguiendo el diseño establecido, se procede a realizar distintas pruebas con la finalidad de corregir errores de diseño y construcción del producto.

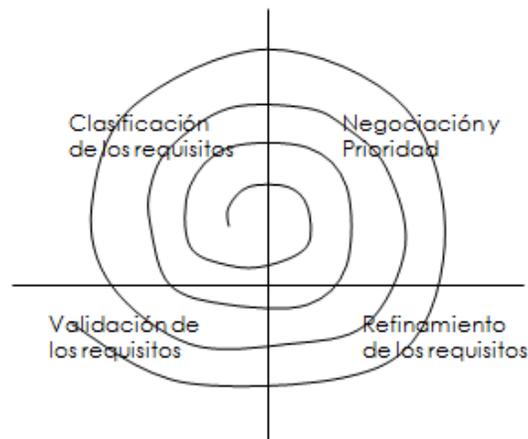


Figura 2.5: Proceso espiral para el análisis de los requisitos.

Aprovisionamiento del componente: especificación de los componentes de *software* existentes que pueden ser reutilizados en el desarrollo del gestor de información.

Pruebas del componente: esta fase tiene como finalidad encontrar los aspectos del componente que no satisfagan los requerimientos, que se cumplan con las restricciones y que el comportamiento sea el deseado.

Certificación del componente: esta fase corresponde a la verificación de todos los aspectos relacionados con el componente, para ello deben cumplirse una serie de pasos y verificar si los cumple. Estos pasos serán explicados en el siguiente capítulo.

Liberación del componente: es la última fase en la creación de un componente de *software* para completar el ciclo de reloj, esta fase se refiere a cuando el componente estará a disposición del público.

Una de las principales ventajas de la metodología *Watch* es que es cíclico, iterativo y controlable, lo cual permite con cada ciclo tener una versión más avanzada del sistema, logrando así mejorar el producto, introducir nuevos requisitos y corregir errores.

2.4. XML

Según W3C (2003) “Extensible Markup Language (XML) es un formato de texto muy flexible derivado de SGML (ISO 8879). Originalmente diseñado para afrontar los retos de las publicaciones electrónicas en gran escala, XML también está desempeñando un papel cada vez más importante en el intercambio de una amplia variedad de datos en la web”.

Según Wikipedia (2009) “XML por sus siglas en inglés de Extensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML)”.

Según Ray (2001) “XML es un lenguaje de marcas, protocolo de contenedores y gestión de la información, también es una filosofía para la información que se maneja, busca la utilidad máxima y flexibilidad para datos refinándolo a su forma más pura y estructurada.”

Una forma bastante sencilla para entender por ejemplo cuando se observa la estructura del periódico se puede distinguir cada uno de sus artículos, la diferencia entre las fuentes utilizada para los títulos entre otras características. Sucede lo mismo para los documentos XML solo que aquí las etiquetas y símbolos determinan la estructura del documento. Un ejemplo bastante sencillo de como está formado un documento XML, es el mostrado en la figura 2.6.

```
<message>  
  <exclamation>Hello, world!</exclamation>  
  <paragraph>XML is <emphasis>fun</emphasis> and  
    <emphasis>easy</emphasis> to use.  
  <graphic fileref="smiley_face.pict"/></paragraph>  
</message>
```

Figura 2.6: Código XML

2.4.1. Ventajas de XML

Ray (2001) afirma que algunas de las ventajas más importantes de los XML son:

- XML puede almacenar y organizar aproximadamente cualquier clase de la información.
- Es un estándar abierto, XML no está atado a ninguna compañía, ni a ningún *software* en particular.
- Unicode es su conjunto de caracteres estándar, XML puede escribir en diversos idiomas con diferentes formas de escritura y símbolos como los chinos.
- XML ofrece muchos modos de comprobar la calidad de un documento, con reglas para la sintaxis, relación interna la comprobación, entre otros.
- Con una sintaxis clara y simple, XML es fácil a leer y analizar tanto por las personas como por el *software* diseñado para esto.

2.5. RSS

Las fuentes RSS y Atom son fuentes para la redifusión web o vulgarmente conocido como fuentes con contenido de sindicación, esto permite que sitios *webs* compartan su contenido con otras aplicaciones de un modo estándar en lenguaje XML como se muestra en el código de la figura 2.7 . Este formato permite que la distribución del contenido se realice sin necesidad de un navegador web de forma rápida y actualizada (Hammersley 2005).

Las fuentes o feed no son más que los titulares y enlaces para mostrar el contenido entre sitios *web*.

Originalmente las fuentes de redifusión fueron creadas y son utilizadas para compartir datos entre sitios *web*, pero con el pasar del tiempo y mejoras en el desarrollo de las fuentes, estas son utilizadas entre sitios *webs* y aplicaciones de escritorio llamados lectores de RSS (Esteban 2006).

El contenido de redifusión permite al usuario obtener actualizaciones en tiempo real sobre los sitios *web* asociados al servicio de fuentes RSS.

```
<item>
    <title><                ></title>
    <link><                 ></link>
    <description><         ></description>
    <author><              ></author>
    <pubDate><            ></pubDate>
</item>
```

Figura 2.7: Código fuente del estándar de los RSS

En la actualidad existen tres tipos de formato RSS y sus siglas adquieren un significado diferente según la especificación usada:

- *Rich Site Summary* (RSS 0.91)
- *RDF Site Summary* (RSS 0.9 y 1.0)
- *Really Simple Syndication* (RSS 2.0)

Los archivos RSS son un nuevo método para obtener y ofrecer información gracias a que contienen metadatos sobre las fuentes de información.

Es sencillo reconocer entre los diversos sitios e informaciones que existen en Internet aquellos que disponen de formato RSS, puesto que los *feeds* suelen indicarse en las páginas *web* con el siguiente icono (figura 2.8).



Figura 2.8: Icono de identificación de los RSS

2.5.1. Utilidad de los RSS

A continuación se presentan las principales razones para las cuales son útiles la redifusión de contenido según Hammersley (2005):

- Es de gran utilidad para sitios *web* que actualicen sus contenidos con frecuencia, ya que permite compartir la información y verla en otros sitios de forma inmediata
- Los RSS no sólo le sirve al usuario para recibir la información, también son utilizados para mostrar los contenidos novedosos de su *web* a otros usuarios.
- Aumenta el tráfico del sitio *web*.
- Cualquier usuario puede suscribirse a una fuente RSS y obtener las últimas noticias a través de un lector de fuentes de redifusión.
- Ayuda a la clasificación por medio de los motores de búsqueda.
- Puede compartir datos y aumentar las relaciones entre comunidades de sitios *web*.
- Mejora la relación del sitio *web* y el usuario.
- Con tecnologías adicionales, permite que otros usuarios puedan hacer notificaciones vía mensajería instantánea.
- Privacidad ya que no usa el correo electrónico para suscribirse a las fuentes de RSS.
- Protección contra el *spam* y los virus ya que es imposible recibir *spam* con los RSS.
- Fácil de cancelar si ya no se quiere recibir un *feed*.

2.6. Síntesis y reconocimiento de voz

Durante los últimos años han surgido interfaces H-C que combinan varias tecnologías del lengua para permitir el acceso y transferencia de información a través del habla. El reconocimiento de voz es el proceso de transformar una señal a texto y la síntesis de voz el proceso de transformar el texto a una secuencia de sonidos. A continuación se explica cada una de estas tecnologías.

2.6.1. Sintetizador de voz

La síntesis de voz es el proceso mediante el cual los computadores y otros dispositivos electrónicos, producen un sonido que se asemeja a la voz de los seres humanos (Parsons 2008).

En la actualidad la síntesis de la voz es una tecnología bastante común en las compañías telefónicas, la mensajería de voz y atención al cliente son un sintetizador que guía al usuario a través de su interfaz auditiva.

También los sintetizadores sirven para leer en voz alta lo que está en las pantallas de los computadores, lo cual es de gran ayuda para las personas con discapacidad visual ya que les permite tener un mejor acceso a los computadores, dispositivos electrónicos hasta a Internet. *Orca*, *Festival*, *Kmouth* son software sintetizadores de voz bajo licencias libres.

Los sintetizadores de voz reúnen fonemas para poder formar las palabras, los fonemas son la unidad de sonido básica, por ejemplo, “veo” “gas”. La voz también puede ser generada artificialmente por un computador (Parsons 2008).

Para lograr la más completa claridad en el habla, lo adecuado es almacenar pronunciaciones de palabras completas.

Un sintetizador de voz básico está formado por un *software* de texto a voz, el cual genera sonidos que son reproducidos por la tarjeta de sonido estándar del computador, también existen sintetizadores que son dispositivos de *hardware* especiales.

2.6.2. Reconocedor de voz

El reconocimiento de voz forma parte de la inteligencia artificial, que tiene como objetivo permitir la comunicación hablada entre seres humanos y computadores (interacción humano-computador). Según Sanders (1992), los sistemas de reconocimiento de voz se basan en el siguiente principio para su funcionamiento, se utiliza un micrófono para convertir el lenguaje humano en señales eléctricas, los patrones de estas señales se transmiten a un computador donde se comparan con un diccionario de patrones que se almacenaron previamente, cuando se encuentra una correspondencia razonable la palabra se reconoce y el computador produce una salida apropiada. Existen dos tipos de sistemas de reconocimiento de voz:

Sistemas dependientes del hablante: los usuarios del sistema tienen que repetir primero varias veces una palabra con el fin de adiestrar al computador para que reconozca su patrón de voz específico.

Sistemas independientes del hablante: los avances en este ámbito se han desarrollado sistemas que pueden reconocer palabras sin importar que usuario las pronuncie.

Los vocabularios de todos los sistemas actuales son muy limitados, los usuarios deben hablar con claridad y hacer una pausa entre las palabras o frases cortas logrando así que el sistema pueda hacer su trabajo. Algunos de los problemas más resaltantes para el reconocimiento de voz se deben al acento del individuo que habla, los dialectos (patrones de voz de la gente que son tan únicos como sus huellas digitales) y al hecho que el significado de las palabras puede variar según el contexto que se usan.

2.6.3. Componentes de *software* para la síntesis y reconocimiento de voz

A continuación se presentan algunos de los principales componentes de software existentes hoy en día para la síntesis y reconocimiento de voz (ver figura 2.9). Estos componentes han sido desarrollados tanto en software libre como software propietario.

- *Dragon naturally speaking*, reconocedor y sintetizador de voz. Desarrollado por *Nuance*.
- *Via voice*, reconocedor y sintetizador de voz. Desarrollado por *IBM*.
- *Fonix speech*, reconocedor y sintetizador de voz. Desarrollado por *Fonix speech, Inc.*
- *Festival*, sintetizador de voz. Desarrollado por *The Centre for Speech Technology Research (CSTR)* de la Universidad de *Edinburgh*.
- *Kmouth*, sintetizador de voz. Desarrollado para ambientes KDE Linux.
- *Orca*, sintetizador de voz. Desarrollado por *Sun Microsystems, Inc.*



Figura 2.9: Componentes de *software* para la síntesis y reconocimiento de voz.

Capítulo 3

Gestor de información, especificación, requerimientos, diseño y aprovisionamiento

www.bdigital.ula.ve

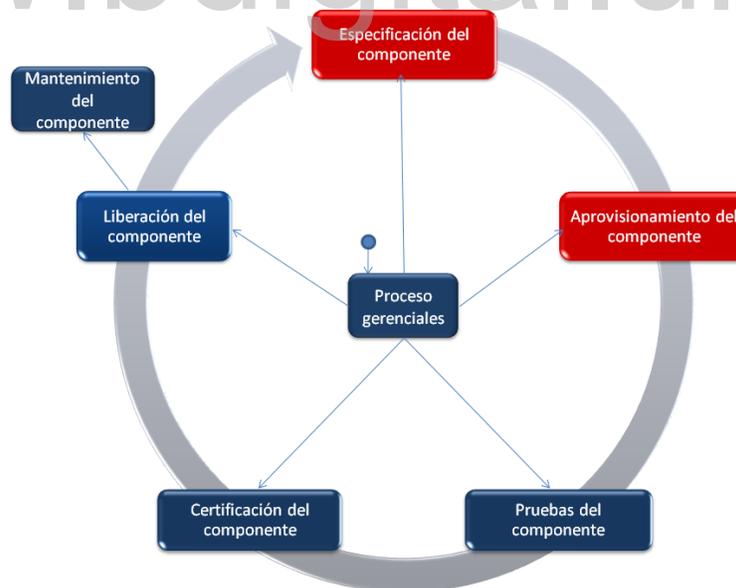


Figura 3.1: Método *Watch Component*, especificación, requerimientos, diseño y aprovisionamiento.

El desarrollo de este capítulo abarca los dos primeros pasos del método *Watch Component*, especificación del componente (donde se incluye la ingeniería de requisitos y diseño arquitectónico) y aprovisionamiento del componente (figura 3.1).

Esta etapa es fundamental para el desarrollo, ya que a través de cada paso se crean bases, restricciones y estructuras, que servirán de guía para la realización de las siguientes etapas.

3.1. Especificación del producto de *software*

El gestor de información es una herramienta que facilita al usuario el servicio de la información en sitios *webs* con contenidos RSS, dedicados a mantener actualizados frecuentemente los sitios *webs*, como páginas de noticias más relevantes del acontecer mundial en sus diferentes ámbitos, foros, *blogs* y redes sociales.

A través de su interfaz auditiva el gestor brinda gran ayuda en la interacción humano-computador (IHC) para usuarios con discapacidades, en particular personas con problemas de visión. No solo responde de forma eficaz a las búsquedas solicitadas, también tiene opción de repetir los titulares encontrados, pasar al siguiente titular o salir de la búsqueda.

El siguiente diagrama (figura 3.2) muestra de una forma sencilla el funcionamiento del gestor basado en captar, recompilar, gestionar y finalmente mantener informado al usuario al tiempo que realiza búsquedas de su interés.

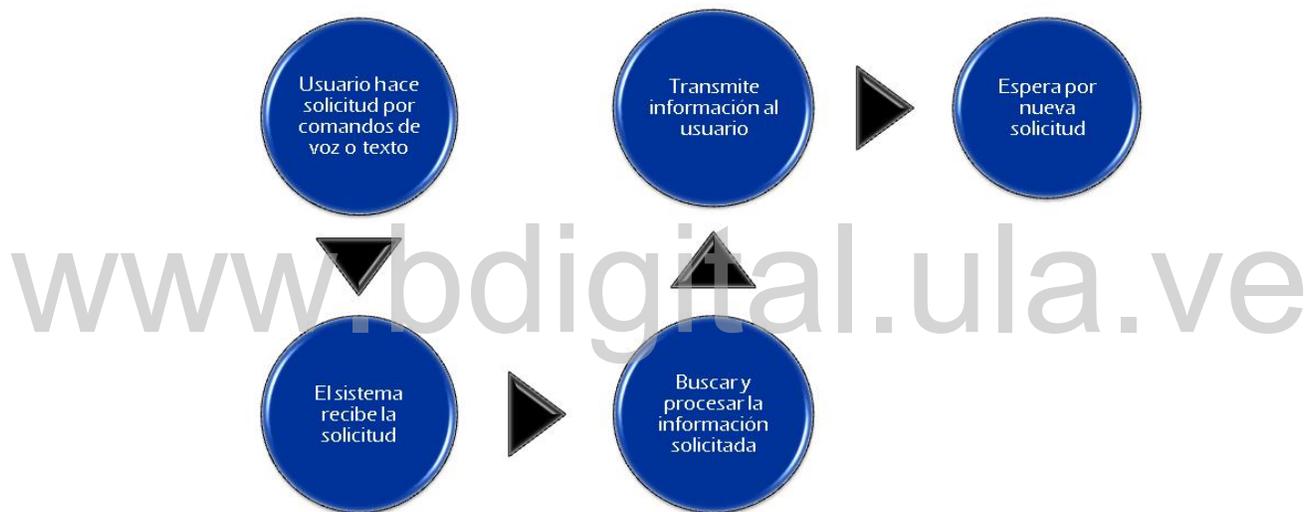


Figura 3.2: Diagrama de funcionamiento del gestor de información

3.2. Análisis y especificación de requerimientos

El análisis y especificación de los requisitos son una parte fundamental en el desarrollo del producto de *software*, puesto que ayuda a entender mejor el problema y a encontrar una solución adecuada, estos indican los objetivos y especificaciones que tiene que cumplir el producto.

A continuación se presentan los actores que están involucrados en el proceso de desarrollo y funcionamiento de la aplicación (figura 3.3); ellos son piezas claves para lograr la captura de los diferentes requisitos del sistema.

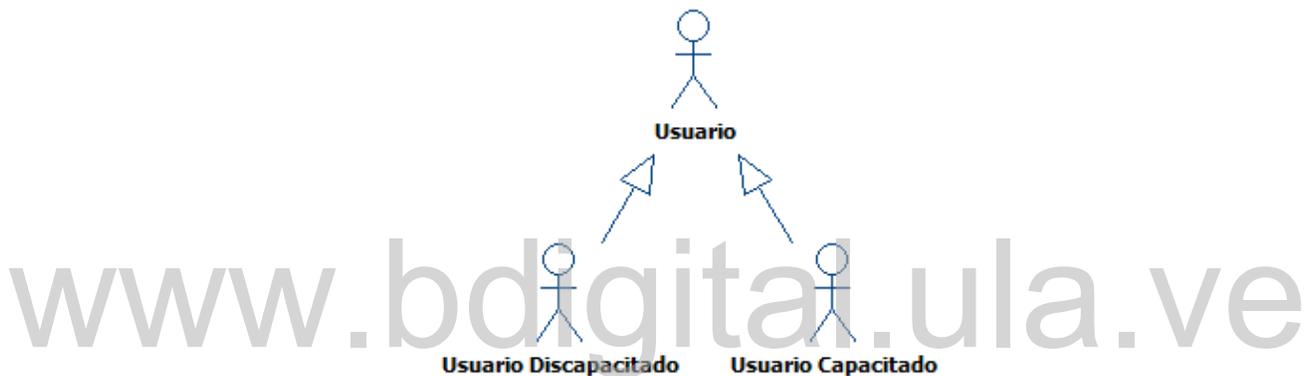


Figura 3.3: Actores del sistema.

Usuario: persona a quien está dirigida la herramienta de *software*, en este caso se tienen dos tipos de usuarios, las personas con capacidades limitadas (personas con discapacidad de visión) y personas sin discapacidades.

En la tabla 3.1 describe los requisitos esenciales para el desarrollo del gestor de información (las tablas de especificación de cada uno de los requisitos se encuentran en el apéndice C), éstos han sido divididos en las categorías accesibilidad, funcionalidad, configuración y *hardware*. También se ha tomado en cuenta si se trata de un requisito funcional o no funcional.

Los requisitos funcionales están asociados con el funcionamiento y comportamiento del sistema, mientras que los requisitos no funcionales son los que imponen restricciones y atributos al producto de *software*.

Para este proyecto se asignan puntajes para priorizar los distintos requisitos del componente de *software* (es importante clasificar los requisitos asignándoles prioridad en la etapa de desarrollo, según mi criterio, una buena forma para el manejo de la prioridades es el siguiente, se tiene un puntaje del 1 al 3 donde la prioridad más importante o alta es 1 y la más baja 3).

Categoría	Numero del requerimiento	Descripción	Calificación	Prioridad
Accesibilidad	1	Fácil acceso y manipulación	No funcional	1
	2	Tener una interfaz auditiva	No funcional	2
	3	Dar las noticias a través de un sintetizador de voz	Funcional	2
	4	Recibir ordenes a través de comandos de voz (reconocedor de voz)	Funcional	2
	5	Presentar una interfaz grafica agradable para el usuario	No funcional	2
	6	Estar orientado a personas con discapacidades de visión	No funcional	2
Funcionalidad	7	Poder hacer búsquedas de múltiples paginas	Funcional	1
	8	Poder descargar de manera rápida y sin presentar perdidas de la información de las paginas con fuentes RSS	Funcional	1
	9	Ir al titular anterior	Funcional	2
	10	Pasar a la siguiente titular	Funcional	2
Configuración	11	Configuración sencilla	Funcional	1
	12	Poder agregar, modificar y eliminar paginas con contenidos RSS	Funcional	1
Hardware	13	Ser una nueva habilidad para el robot personal Maggie	No funcional	1
	14	Tener acceso a internet	No funcional	1
	15	Tener periféricos multimedia (micrófono y parlantes)	No funcional	1

Tabla 3.1: Tabla de requisitos del gestor de información.

3.2.1. Casos de uso

A continuación se describen los casos de usos obtenidos a partir de los requisitos del gestor de información. Los casos de uso se han dispuesto según una jerarquía piramidal donde se tiene en su cúspide el diagrama que describe el uso general del gestor y en su base la descomposición de cada caso de uso.

Diagrama, buscar información y configurar las fuentes RSS: En la figura 3.4 se muestra el diagrama que representa el principal requerimiento y funcionalidad del gestor de información. Se puede observar las actividades que pueden ejecutar los usuarios en la aplicación, realizar búsqueda de noticias de fuentes RSS previamente configuradas o configurar dichas fuentes, en los siguientes diagramas estos casos son desglosados y explicados con mayor profundidad.

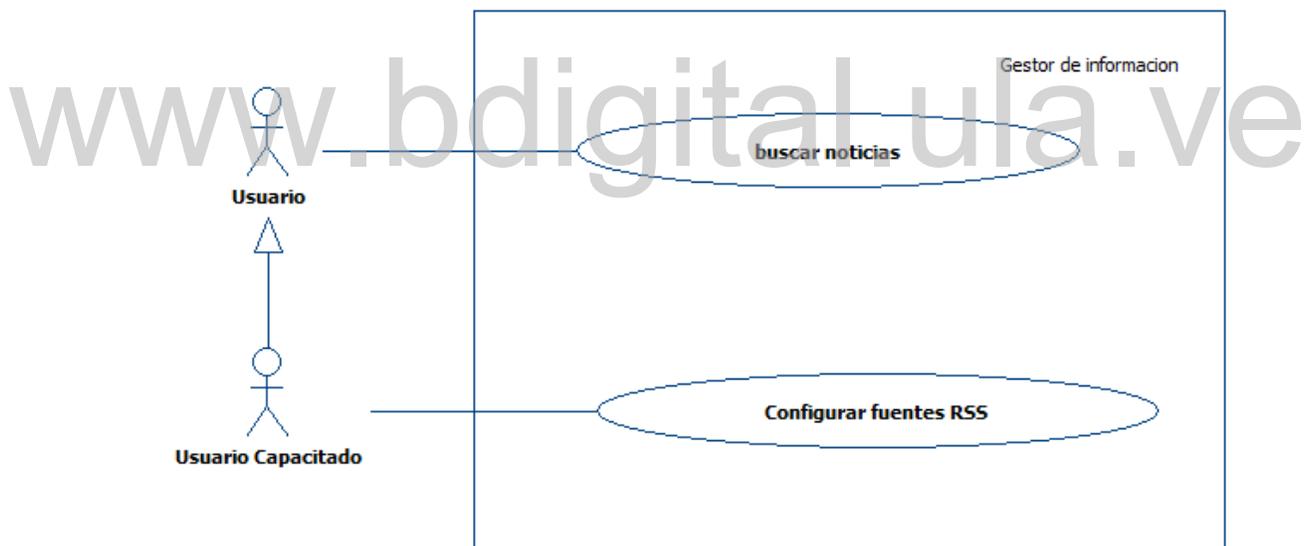


Figura 3.4: Diagrama del uso general del gestor de información.

Diagrama del caso de uso buscar noticias: En la figura 3.5 del diagrama referente a la búsqueda de noticias, el usuario suministra información necesaria para realizar una búsqueda, esto desencadena una serie de acciones que debe ejecutar la aplicación. Verificar que existan URLs de la categoría que se desea buscar, descargar de dichas URLs la información y finalmente procesarla para transmitir las noticias este caso de uso se describe mejor en el siguiente diagrama.

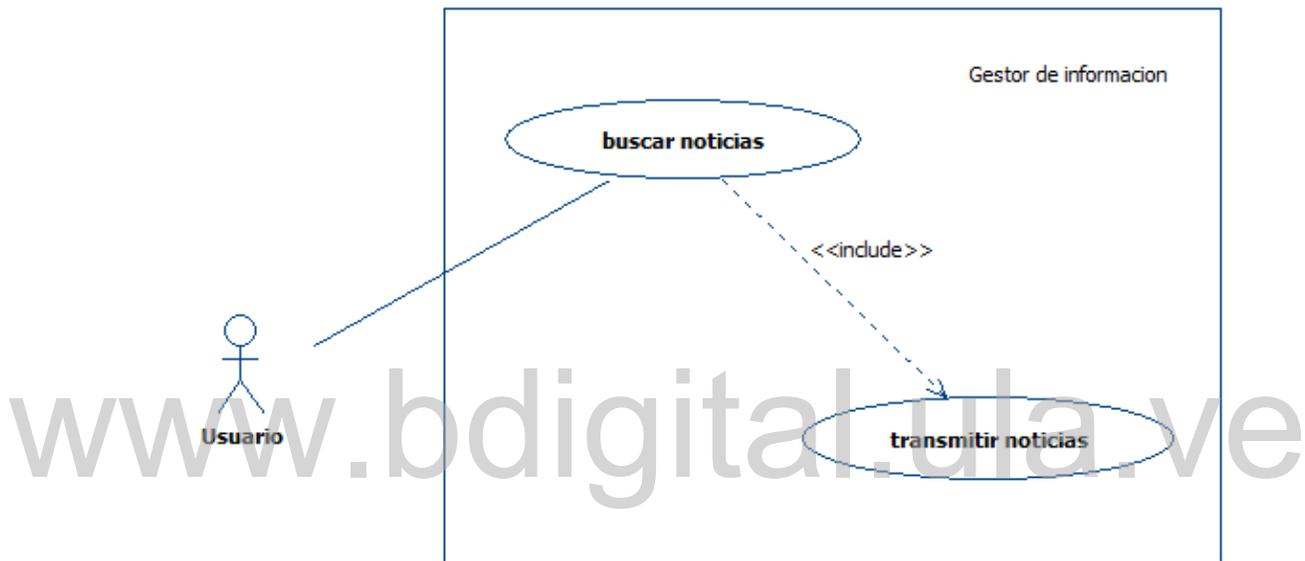


Figura 3.5: Diagrama caso de uso buscar noticias.

Diagrama del caso de uso transmitir noticias: En este caso de uso se observa (figura 3.6) las diferentes opciones que puede accionar el usuario mientras el gestor de información transmite las noticias a través de su interfaz auditiva. El usuario tiene la posibilidad de accionar los siguientes casos, pasar al siguiente titular, repetir el titular previamente dicho o detener la aplicación (estas acciones son tratadas con mayor detalle en los diagramas de actividades y secuencias). En la tabla de especificación (ver tabla 3.2) de este caso de uso se observa cuales son las descripción, precondiciones y postcondiciones para este caso de uso.

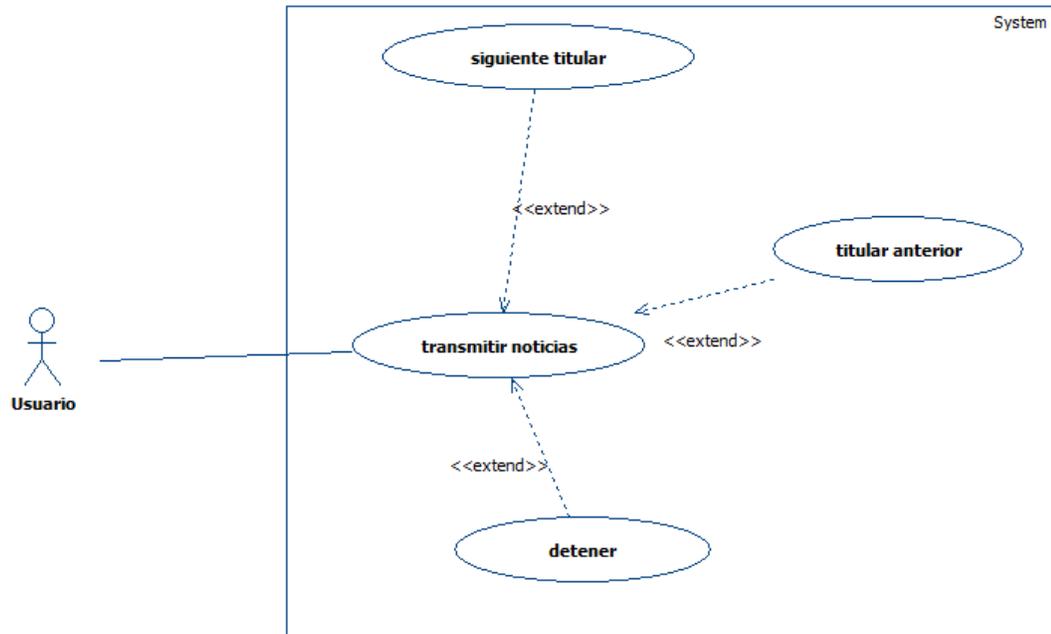


Figura 3.6: Diagrama caso de uso transmitir noticias.

Caso de uso	
Nombre	Transmitir noticias
Descripción	Luego de realizar la búsqueda el gestor brinda algunas opciones al usuario mientras lee las noticias encontradas, como es el caso de repetir el titular, pasar al siguiente titular o detener la aplicación.
Pre condición	Haber realizado una búsqueda de noticias exitosa.
Post condición	A través de la interfaz auditiva el sistema inicia la transmisión de las noticias al usuario
Comportamiento	La aplicación está a la espera ya que el usuario puede ordenarle al repetir el titular previamente dicho, pasar al siguiente titular o detener la búsqueda.

Tabla 3.2: Especificación, caso de uso transmitir noticias.

Diagrama del caso de uso configurar las fuentes RSS: En el diagrama de la figura 3.7 describe el caso de uso configurar las fuentes RSS, en él se puede observar que existe una serie de casos de uso que el usuario puede disparar dependiendo de lo que él quiere ejecutar, para tener una mejor visión de cada uno de estos casos en la figura 3.8 muestra los diferentes flujos alternativos que pueden ocurrir dependiendo de cada elección (estos flujos alternativos se pueden observar con mayor claridad en la captura de los diseños de pantallas en las próximas secciones).

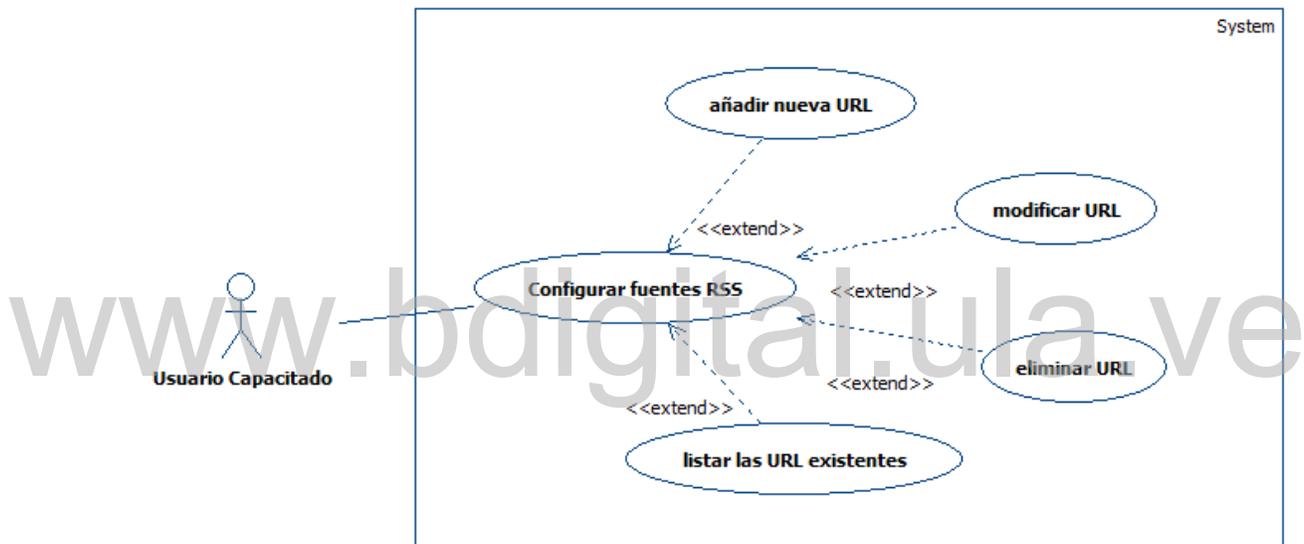


Figura 3.7: Diagrama caso de uso configurar las fuentes RSS.

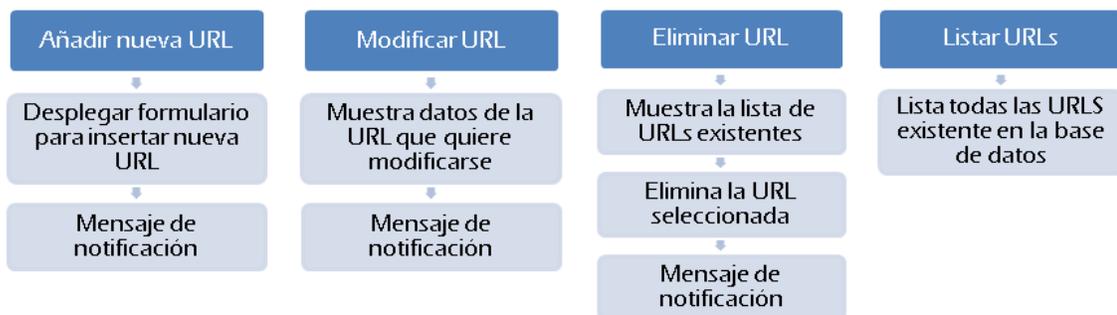


Figura 3.8: Diagrama de flujos alternativos del caso de uso configurar las fuentes RSS.

La tabla 3.3 la especificación del caso de uso, durante esta fase el usuario capacitado es quien puede realizar la configuración de las URLs para las futuras búsquedas que realizara el gestor de información, allí el usuario puede añadir, modificar, listar o eliminar páginas webs.

Caso de uso	
Nombre	<i>Configurar las fuentes RSS</i>
Descripción	<i>A continuación se presenta el comportamiento de la aplicación cuando el usuario quiere añadir una nueva dirección URL modificar algún URL existente, listar o simplemente eliminar alguna fuente de la base de datos.</i>
Pre condición	<i>Haber activado la opción de configuración del gestor de información.</i>
Post condición	<i>Informar al usuario si la configuración fue exitosa.</i>
Comportamiento	<i>El gestor de información espera que el usuario ejecute una de las opciones de configuración de los sitios webs con contenido RSS y responde adecuadamente modificando la base de datos del gestor de información.</i>

Tabla 3.3: Especificación, caso de uso configurar las fuentes RSS.

3.2.2. Diagramas de actividades y diagramas de secuencias

Los diagramas de actividades sirven para modelar un conjunto de actividades y el flujo de trabajo entre ellos. Mientras los diagramas de secuencia muestran la interacción entre un conjunto de clases, como debe ejecutar las acciones y describen el comportamiento por la acción del usuario durante el funcionamiento.

La figura 3.9 representa el diagrama de actividades de la búsqueda de información, en el se aprecia claramente cuál es el conjunto de actividades que debe realizar el componente de software para dicha búsqueda. El usuario solicita una nueva búsqueda y el gestor evalúa si dicha información está disponible, de ser así comienza la descarga, la procesa y comienza la transmisión de los titulares al usuario. En este punto el usuario tiene la posibilidad pedirle al gestor que repita la información del titular previamente dicha, pasar al siguiente titular o finalizar la búsqueda. Si la información solicitada por el usuario no está disponible, puede realizar una nueva búsqueda.

En la figura 3.10 se muestra la ejecución de una búsqueda exitosa de la aplicación desde el momento en que se levanta el constructor, siguiendo con la orden del usuario, luego recuperando, procesando y ejecutando una serie de pasos para su procesamiento hasta que finalmente es transmitida la información al usuario.

Durante la transmisión de los resultados el usuario puede solicitar ir a la noticia que anteriormente se ha mencionado (ver Anexo figura C.1).

El diagrama representado en el apéndice C figura C.2 muestra cual es la ejecución que debe realizarse para ir al siguiente titular y el diagrama del apéndice C figura C.3 detener el gestor de información. Estos tres diagramas describen el flujo feliz cuando se realiza una búsqueda de noticias siguiendo con el desarrollo realizado en los diagramas de casos de uso y secuencia.

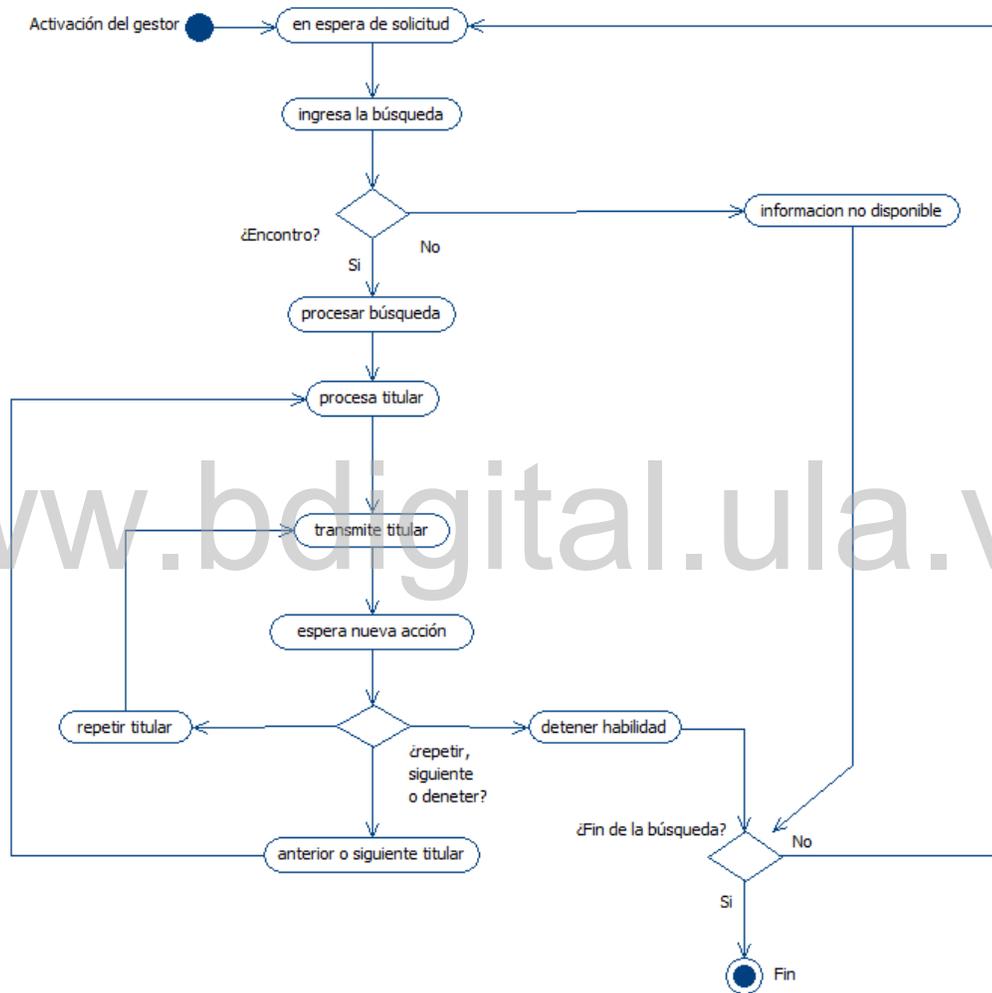


Figura 3.9: Diagrama de actividad búsqueda de información.

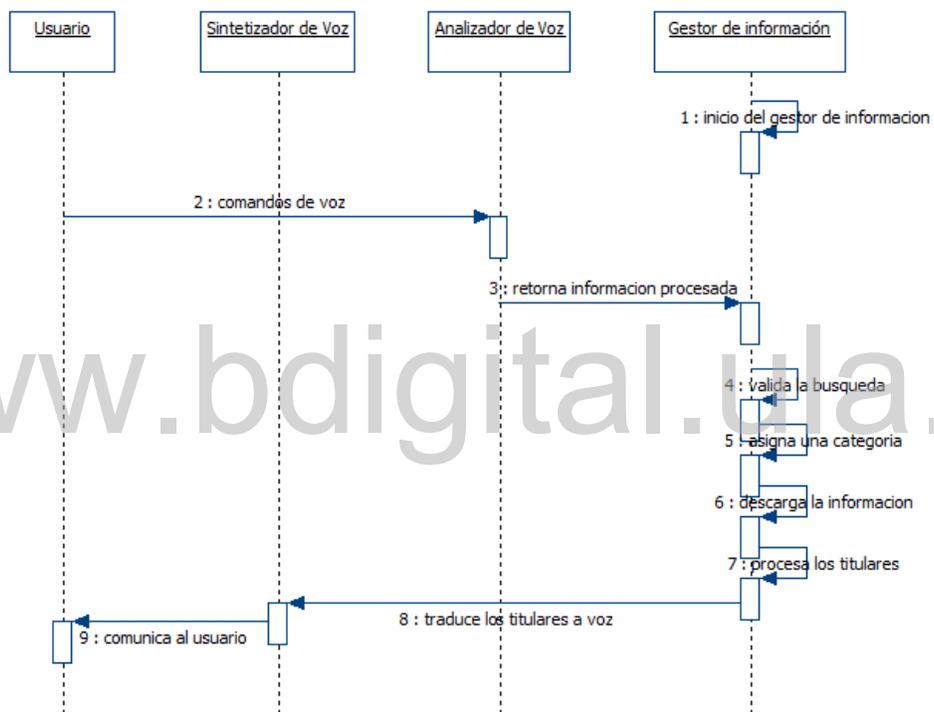


Figura 3.10: Diagrama de secuencia búsqueda exitosa.

3.3. Diagrama de clases y entidad relación

Los diagramas de clases son parte importante del diseño del *software*, ya que describen la estructura del sistema.

A continuación se describe el diagrama conceptual de clases (figura 3.11), las vistas de implantación de los diagramas de clases (ver apéndice D figura D.1) y entidad relación (ver figura 3.12) para el gestor de información. Seguidamente se explica paso a paso cada clase, sus respectivos miembros y funciones:

Clase Gestor Información (ver apéndice D tabla D.2) depende de varios componentes de *software* importantes para su funcionamiento, Loquendo-ASR es el analizador de voz que es el encargado en captar la voz del usuario el cual puede ejecutar un conjunto de ordenes durante la ejecución de la herramienta de *software*, Festival es un sintetizador de voz, quien da las noticias de las diversas búsquedas que quiera hacer el usuario. Las clases *mysql++* y *cURL* facilitan la conexión con la base de datos del sistema, el procesamiento de los documentos XML y su descarga de la red respectivamente.

Configuración *URL* (ver apéndice D tabla D.1) es una clase asociadas a Gestor Información, están conectadas a través de *url*. La clase configuración ejecuta la configuración del sistema, está conectada directamente a la base de datos del sistema. Siguiendo con la descripción de la clase Gestor Información se tienen un conjunto de funciones que se describen a continuación:

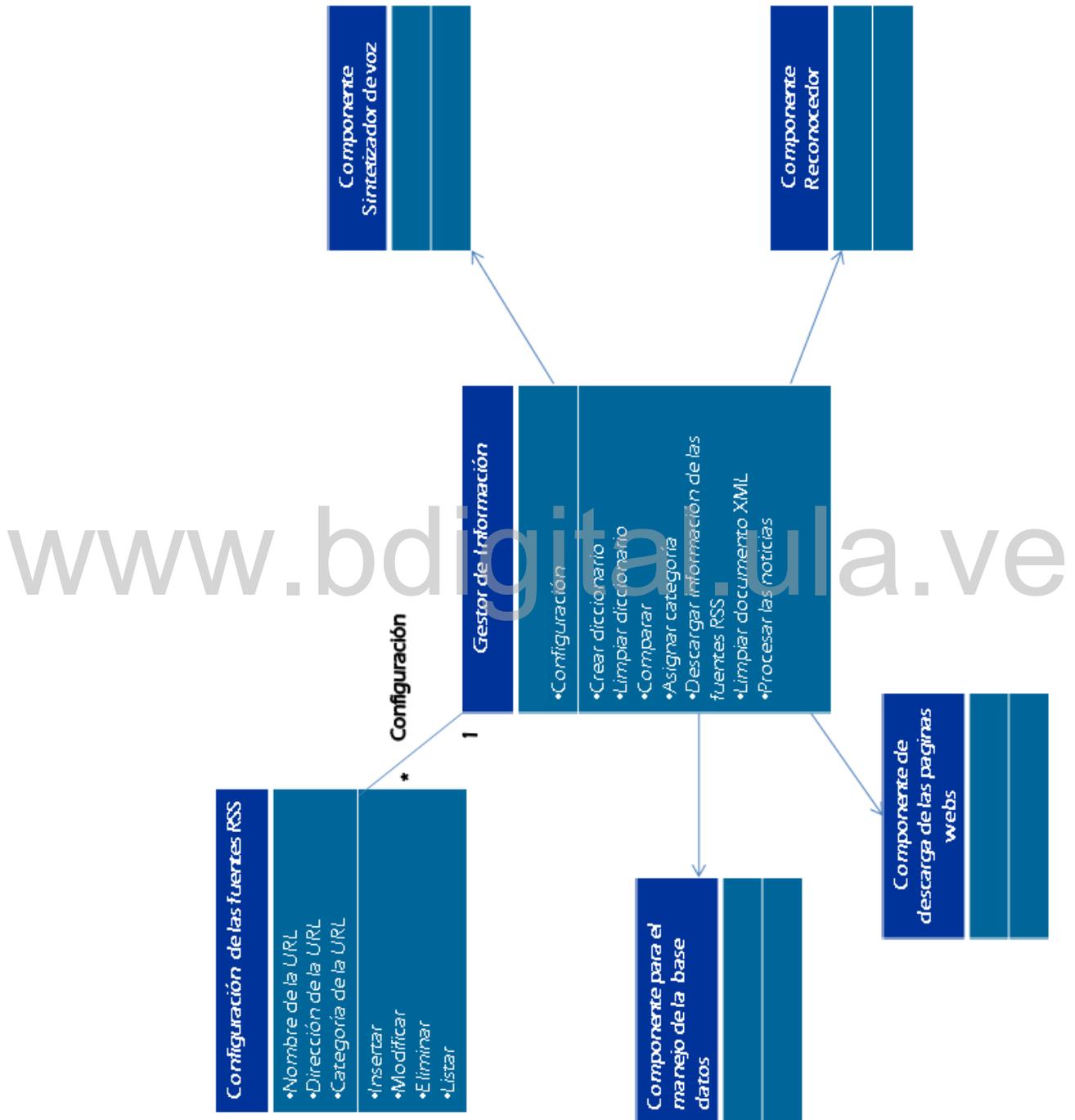


Figura 3.11: Diagrama conceptual de clases.

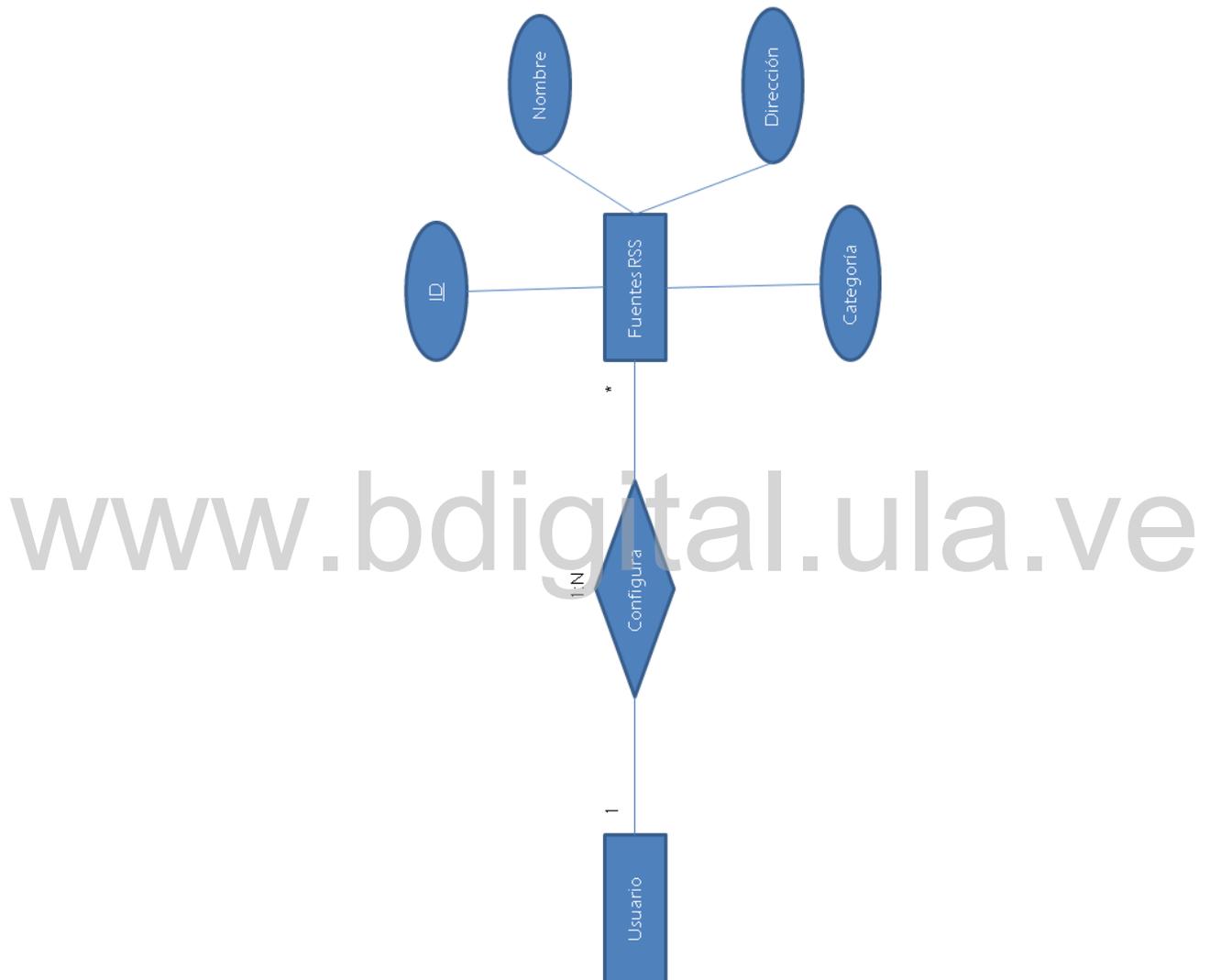


Figura 3.12: Diagrama entidad relación.

3.4. Diseño del producto de *software*

El diseño del *software* se presenta de dos maneras, primero debe representarse la arquitectura del sistema, donde se muestran las estructuras de datos y los componentes del programa necesarios para construir un sistema computacional.

Define el estilo arquitectónico que tomará el sistema, la estructura y las propiedades de los componentes que constituyen el sistema e interacciones entre los componentes del sistema. Luego se modelan las interfaces que conectan el *software* con los usuarios finales, con otros sistemas, dispositivos y con los propios componentes que lo constituyen.

Se identifican los objetos, acciones de la interfaz y luego se crea un formato de pantalla que forma la base de un prototipo de interfaz de usuario, dicha interfaz usuario crea un medio de comunicación entre el ser humano y el computador.

También se diseñan los componentes de *software* que se utilizarán en la construcción del gestor de información, definiendo las estructuras de datos, los algoritmos, las características de la interfaz y los mecanismos de comunicación asignados a cada componente de *software*.

3.4.1. Diseño de la arquitectura del sistema

En la figura 3.13 describe conceptualmente la arquitectura general del sistema. Este diagrama de bloques facilita la comprensión del funcionamiento del gestor de información.

Usuario: interactúa de forma directa con el gestor de información solicitando y recibiendo información, por medio de dos tipos de interfaces auditiva y grafica; estas interfaces sirven de gran ayuda al usuario discapacitado.

Gestor de información: es el principal componente del sistema ya que es aquí donde se ejecutan las acciones de todo el proceso de búsqueda (motor de búsqueda), siendo este componente la mente de todo el sistema.

Almacenamiento feed RSS: el sistema necesita de una pequeña base de datos para el manejo de los sitios *web* asociados a las búsquedas. Esta base de datos almacena

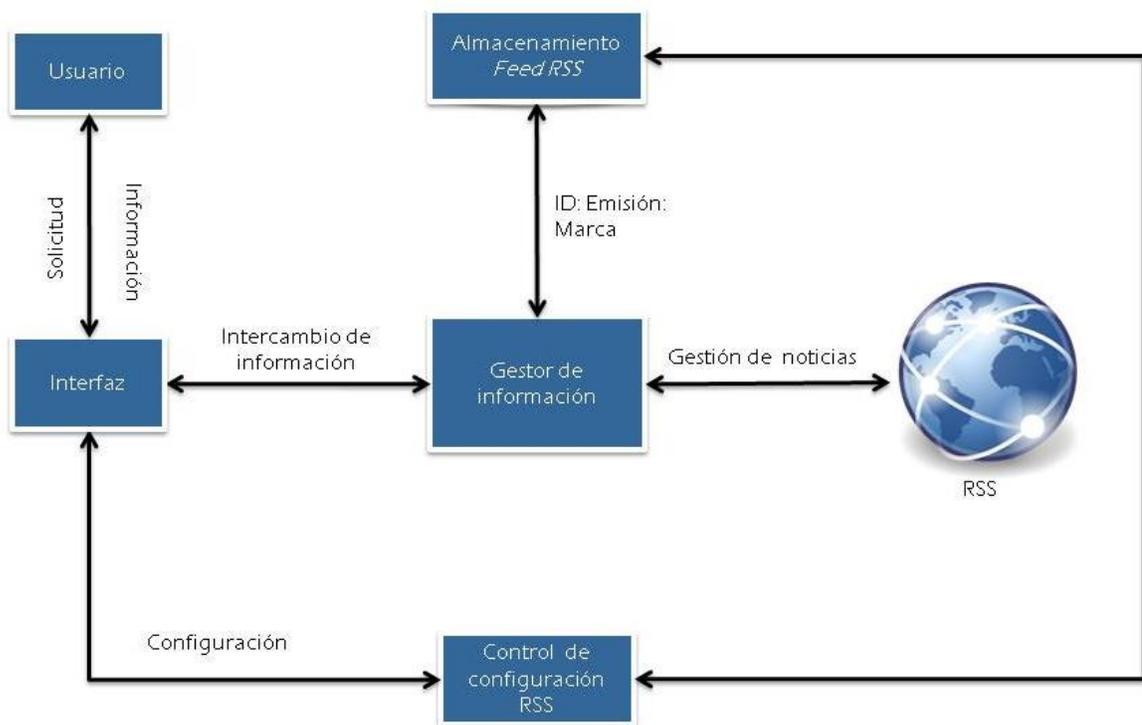


Figura 3.13: Diagrama de bloques, arquitectura general del sistema.

la información necesaria para realizar la gestión de la información (nombre, dirección *web* y categoría de las *URLs*). Utilizando un campo clave el gestor realiza la búsqueda de información de los sitios *web* correspondientes a la solicitud del usuario.

Control de configuración RSS: a través de la configuración del sistema el usuario puede hacer el manejo de base de datos del gestor agregando, modificando las fuentes de RSS de su interés. Esto le otorga libertad al usuario sobre la información que desea manejar en cada una de sus búsquedas.

El diseño de componentes está basado en una arquitectura de una capa (figura 3.14). Para la construcción del prototipo se requieren los siguientes componentes que han sido descritos previamente: sintetizador de voz, reconocedor de voz, analizador sintáctico, descarga de fuentes RSS y manejador de la base de datos.

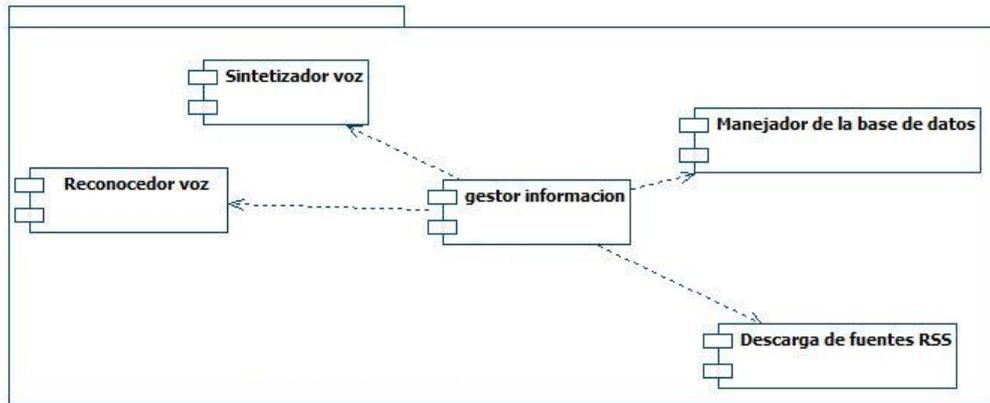


Figura 3.14: Diagrama de componentes, arquitectura del sistema.

Sintetizador de voz: este componente está encargado de prestar el mayor servicio de accesibilidad a las personas con discapacidad ya que es el encargado de transmitir las noticias encontradas al usuario.

Reconocedor de voz: al igual que el componente de la síntesis de voz da accesibilidad, el reconocedor le da al usuario invidente la capacidad de poder realizar búsquedas en Internet sin necesidad de utilizar el teclado, simplemente utiliza un micrófono para descargar información de páginas *webs* previamente configuradas.

Descarga de fuentes RSS: este componente es quien realiza la descarga de información de las páginas *webs*, pieza clave en el motor de búsqueda del gestor de información.

Manejador de la base de datos: encargado de la conexión entre la base de datos de fuentes RSS y el gestor de información. Este componente le permite al gestor de información saber de cual página debe descargar la información.

3.4.2. Diseño de las interfaces auditiva y gráfica del sistema

Durante esta etapa de diseño de las interfaces de audio y gráfica del sistema se tomaron en cuenta el siguiente conjunto de pasos:

- La información desarrollada durante el análisis de los requerimientos y las acciones de la interfaz.
- Definir eventos (acciones del usuario) que cambiaran el estado de la interfaz.
- Representar cada estado de la interfaz gráfica tal como lo verá el usuario del sistema.

Ya creado un prototipo de interfaz usuario operacional, este es evaluado para determinar si satisface las necesidades del usuario.

Los usuarios del gestor de información deben conocer el manejo básico de un computador, por ello se recomienda que personas con discapacidades de visión sean orientadas durante el proceso.

A pesar que el gestor está orientado para personas con discapacidades también puede ser utilizado por cualquier persona de cualquier edad, con lo cual el producto busca garantizar que sea de fácil manipulación.

3.4.3. Diseño de pantallas

Al iniciar el gestor de información se despliega una interfaz *web*, con una pantalla de inicio sencilla (figura 3.15), donde se muestra una barra de búsqueda y acceso a la configuración de la herramienta de *software* (figura 3.16). Dependiendo cual sea la opción que el usuario quiera realizar se ejecutarán las siguientes acciones:

Buscar: escribiendo en la barra de búsqueda o a través del reconocedor de voz (opcional) el usuario podrá hacer búsqueda de sus noticias preferidas, consultar actualizaciones en los foros, *blogs* y redes sociales de su interés.

Configuración: accede a la configuración del sistema, donde el usuario puede gestionar agregar (ver apéndice E figura E.1), modificar (ver apéndice E figura E.2), eliminar (ver apéndice E figura E.3), listar (ver apéndice E figura E.4) las páginas *webs* con fuentes RSS almacenadas en la base de datos.



Figura 3.15: Captura de pantalla, página principal.



Figura 3.16: Captura de pantalla, configuración del sistema.

3.5. Aprovisionamiento de los componentes de *software*

Una vez realizada la especificación de requisitos y el diseño del gestor de información, ya se tiene claro cuáles son los componentes necesarios para la construcción del *software*.

La tercera fase del método *Watch Component* abarca el aprovisionamiento y adquisición de componentes, cuales son las características que deben cumplir para ser aceptados, entre otras.

3.5.1. Adquisición de los componentes de *software*

Durante el proceso de adquisición y búsqueda de componentes se encontraron diversos componentes que pueden ser reutilizados, adaptados y útiles para el gestor de información.

Luego de realizar la verificación y revisión de cuales componentes de *software* son los que mejor se adaptan a las necesidades y cumpliendo los requisitos del gestor de información se han obtenido las siguientes tablas, divididas según su funcionalidad (ver tabla 3.4 y tabla 3.5).

Existen gran cantidad de reconocedores y sintetizadores de voz, *Dragon Naturally Speaking* y *Via Voce* son algunos de los más conocidos, estos son paquetes de *software* están en capacidad de realizar tanto el reconocimiento como la síntesis de voz, *Dragon Naturally Speaking* posee versiones *SDK Client* para el desarrollo de aplicaciones bajo *Microsoft visual C++* en sistemas operativos *Microsoft Windows*.

Se optó por el reconocedor de voz *Loquendo* utilizado en el robot *Maggie* en el *Robotics Lab* de la Universidad Carlos III de Madrid, el cual cumple con los requisitos necesarios para el desarrollo de la aplicación, pero este reconocedor presenta una gran limitación ya que su licencia está atada al hardware de *Maggie*. Por esta razón se decidió utilizar *Dragon Naturally Speaking* y sus componentes SDK en la versión 1.0 del gestor de información.

En cuanto respecta al sintetizador de voz *Festival* (desarrollado por *The University of Edinburgh*) es un *software* que se adapta muy bien al gestor, además este sintetizador

Componente	Descripción	Fuente	Ventaja	Desventaja	Aceptación
Dragon Naturally Speaking	Sintetizador y reconocedor de voz	http://www.nuance.com/naturallyspeaking/	Posee tanto reconocimiento como síntesis de voz	Software privativo (costoso) solo puede ser aplicado sobre sistemas operativos Windows, no cuenta con la documentación apropiada, lo que dificulta su utilización	Aceptado para la versión 1.0 DEMO
Festival	Sintetizador de voz	http://www.cstr.ed.ac.uk/projects/festival/	Software libre, presenta la documentación necesaria para su adaptación (bibliotecas y API para c++)	Su pronunciación de las palabras es simple	Aceptado
Orca	Sintetizador y magnificador de voz y braille	http://projects.gnome.org/orca/	Este componente puede trabajar independientemente de la aplicación siempre y cuando cuente con un sistema operativo linux con entorno gnome	Ninguna	Puede ser utilizado perfectamente en el caso que el sintetizador o reconocedor falle
Via Voce	Reconocedor de voz	http://www-01.ibm.com/software/pervasive/embedded/viaoice/	Posee tanto reconocimiento como síntesis de voz	Software privativo (costoso) sin documentación apropiada para ser utilizado como componente.	Rechazado
Loquendo	Reconocedor de voz	http://roboticslab.uc3m.es/roboticslab/	Se adapta a los requisitos del gestor de información (posee la documentación necesaria para ser implantado)	El software esta atado al componente de hardware del robot Maggie	No pudo implementarse

Tabla 3.4: Comparación de los componentes sintetizadores y reconocedores de voz.

Componente	Descripción	Fuente	Ventaja	Desventaja	Aceptación
MiniXML	Herramienta para la manipulación de documentos XML	http://www.minixml.org/	Posee, biblioteca API que se adapta perfectamente a la aplicación.	Solo funciona para paginas codificadas en UTF-8 y UTF-16	Rechazado
CURL	Componente que permite la transferencia y descarga de información en Internet	http://curl.haxx.se/	Es adaptable a múltiples lenguajes de programación y cumple con los requisitos del componente.	Ninguna	Aceptado
Mysql++	Componente para el manejo de la base de datos a través de código escrito en c++	http://tangentsoft.net/mysql++/	Es liviano, fácil de manipular, posee los API necesarios para su manejo en c++.	Ninguna	Aceptado

Tabla 3.5: Comparación de los componentes de *software*.

está creado bajo licencias libres y presenta una biblioteca *API* con el lenguaje *C++*, lo cual permite su utilización sin necesidad de ninguna adaptación al sistema.

CURL(desarrollado por *Haxx AB*) es un componente de *software* que permite la descarga de información de Internet, desarrollado bajo licencias libres y cuenta con múltiples *APIs* para diferentes lenguajes de programación entre ellos *C++*(lenguaje en el cual está desarrollado el gestor de información).

MINIXML (desarrollado por Michael Sweet) herramienta que permite la manipulación de documentos XML a través de sus diferentes funciones, este es un componente esencial para el gestor de información ya que es el encargado del procesamiento del documento XML descargado para poder ser transmitido al usuario a través del sintetizador de voz. El componente es de gran utilidad pero presenta una limitación importante ya que solo funciona para documentos XML codificados en UTF-8 y UTF-16. Por lo que se decidió crear un analizador sintáctico propio que sirva para múltiples tipos de codificaciones de los documentos XML.

Estos componentes escogidos, se adaptan, proporcionan resultados deseados en el gestor de información, además han sido desarrollados en fuentes abiertas (caja blanca) y modularmente, por ende son mucho más sencillos de comprender.

Capítulo 4

Gestor de información, pruebas, certificación y liberación

Siguiendo las fases correspondientes del método *Watch Component* (figura 4.1) pruebas, certificación y liberación del componente son fases de gran importancia en el desarrollo de un producto de *software* ya que en este momento es donde se ponen a prueba las bases, requisitos y estructura definidos en el capítulo anterior.

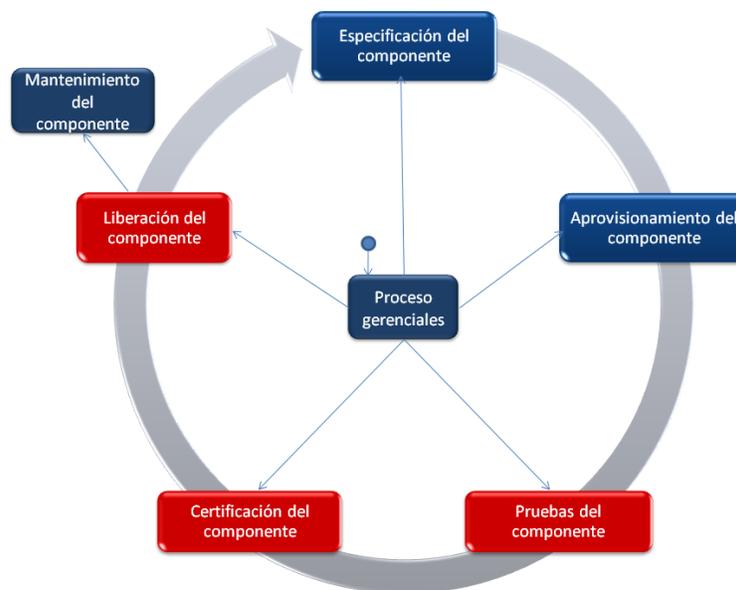


Figura 4.1: Método Watch component

4.1. Pruebas del componente de *software*

Las pruebas están diseñadas con el objetivo de encontrar los aspectos que no satisfagan los requerimientos, cumplan con las restricciones previamente definidas en el capítulo 3 y que el comportamiento del componente sea el deseado.

Una buena prueba es aquella que tiene una alta probabilidad de encontrar errores en el componente.

Durante esta fase, se determinan y aplican las pruebas que garanticen encontrar el mayor número de errores fallas del gestor de información y poder corroborar que los resultados obtenidos del sistema sean los esperados.

Las pruebas son presentadas de la siguiente manera, pruebas de comportamiento de las funciones importantes del sistema (caja negra, prueba de integración), prueba de funcionamiento, prueba de confiabilidad, configuración y de recuperación.

- Pruebas caja negra.
- Pruebas funcionales.
- Pruebas de confiabilidad.
- Pruebas de instalación.
- Prueba de recuperación

4.1.1. Pruebas caja negra

Consiste en aplicar pruebas que demuestre que cada una de las funciones y métodos implantados son plenamente operacionales, desconociendo como funcionan internamente. Simplemente se pasan entradas a las funciones y se comparan las salidas resultantes y las esperadas.

En la tabla 4.1 se observan los resultados obtenidos después de realizar estas pruebas, allí se especifica cuál es la entrada, acción y validación de la prueba.

Función	Valores	Acción	Validación
crear_Diccionario()	Si existe la base de datos de fuentes RSS	Crea diccionario	Valido
	Si no existe la base de datos de fuentes RSS	Ninguna	Invalido
limpiar_Diccionario()	Si diccionario existe	Limpia el diccionario de palabras repetidas	Valido
	Si diccionario no existe	Ninguna	Invalido
comparar(string)	Cadena string = deportes	La función retorna 0, la palabra es encontrada en el diccionario	Valido
	Cadena string = asf123	La función retorna 1, la palabra no se encontró	Valido
asignar_categoria(string)	Cadena string = deportes	Retorna 0, encuentra una categoría a la cual asignar la búsqueda	Valido
info_url()	Siempre que exista un URL	Descarga la información del XML	Valido
limpiar_palabra(string&)	Cadena string = áéíóú	Retorna 0, la palabra se limpio correctamente	Valido
	Cadena string = aeiou	Retorna 0, no hubo necesidad de limpiar la palabra	Valido
procesar_titulares()	Sino existe infoXML.xml	Ninguna	Invalido
	Si existe infoXML.xml	Retorna 0, proceso correctamente el archivo	Valido

Tabla 4.1: Pruebas caja negra para las funciones y métodos del gestor de información

4.1.2. Pruebas funcionales

Durante esta fase de las pruebas, se comprobó el funcionamiento correcto de todo el gestor de información y el cumplimiento de los requisitos propuestos para el desarrollo de la aplicación. Donde trabajan en conjunto todos los componentes de *software* involucrados en el sistema.

A continuación se muestra una de las corridas de esta prueba, asignando valores en la entrada del gestor se recorrió cada una de las fases de todo el proceso y se obtuvo lo siguiente:

Entrada: Buscar las últimas noticias de deportes (figura 4.2).



Gestor de Información

[Configuración](#)

www.bdigital.ula.ve

Figura 4.2: Captura de pantalla, búsqueda.

El reconocedor de palabras recorre la cadena de la entrada en busca de palabras claves para asignar una categoría correcta a la búsqueda comparando con su diccionario interno y encuentra “deportes”.

Luego compara la palabra encontrada con las categorías existentes en la base de datos de fuentes RSS (figura 4.3) y asigna la categoría de búsqueda.

El gestor a través del componente curl descarga la información de la fuente como se muestra en la figura 4.4, luego el documento es procesado y limpiado por las funciones correspondientes (figura 4.5).

Finalmente a través del sintetizador de voz festival se transmiten las noticias de la búsqueda.

Durante la prueba se pudo comprobar el buen funcionamiento del sistema y el cumplimiento de los requisitos propuestos en el capítulo 2 (ver tabla 4.2).

```
mysql> SELECT * FROM url;
+-----+-----+-----+-----+
| id | nombre          | direccion                                     | categoria |
+-----+-----+-----+-----+
| 1 | El mundo        | http://rss.elmundo.es/rss/descarga.htm?data2=59 | portada   |
| 2 | AS DEPORTES    | HTTP://WWW.AS.COM/RSS/FEED.HTML?FEEDID=61 | deportes |
| 3 | Saber ULA      | http://www.saber.ula.ve/feed/rss\_2.0/123456789/5268 | noticias  |
| 4 | Barra Punto    | http://barrapunto.com/index.rss | tecnologia|
| 5 | BBC PORTADA    | http://www.bbc.co.uk/mundo/index.xml | portadas  |
+-----+-----+-----+-----+
```

Figura 4.3: Captura de pantalla, base de datos.

```
</image>
<item>
<title><![CDATA[AarÃ³n reconoce estar "muy desanimado" porque la lesiÃ³n "parece muy grave"]></title>
<link><![CDATA[http://www.as.com/futbol/articulo/aaron-reconoce/dasten/20091111dasdasten\_2/Tes]]></link>
<description><![CDATA["Viendo cÃ³mo estÃ¡ la rodilla soy partidario de pensar en lo peor",
dijo el jugador que se lesionÃ³ ayer ante el Tenerife en la Copa.]]></description>
<author><![CDATA[EFE]]></author>
<pubDate><![CDATA[Wed, 11 Nov 2009 15:53:00 +0100]]></pubDate>
</item>

<item>
<title><![CDATA[Del Potro jubila a Safin]]></title>
<link><![CDATA[http://www.as.com/tenis/articulo/potro-jubila-safin/dasten/20091111dasdasten\_2/Tes]]></link>
<description><![CDATA[El argentino Juan MartÃ³n del Potro derrotÃ³ hoy en la segunda ronda de ParÃ¡s-Bercy
a Marat Safin lo que significa la retirada definitiva del ruso.]]></description>
<author><![CDATA[EFE]]></author>
<pubDate><![CDATA[Wed, 11 Nov 2009 15:43:00 +0100]]></pubDate>
</item>
```

Figura 4.4: Captura de pantalla, infoXML.xml .

```
Aaron reconoce estar "muy desanimado" porque la lesion "parece muy grave"
Viendo como esta la rodilla soy partidario de pensar en lo peor"
dijo el jugador que se lesiono ayer ante el Tenerife en la Copa.
Del Potro jubila a Safin
El argentino Juan Martin del Potro derroto hoy en la segunda ronda de Paris Bercy a Marat Safin
lo que significa la retirada definitiva del ruso.
Undiano Mallenco arbitrara el Barcelona Real Madrid
```

Figura 4.5: Captura de pantalla, documento noticia.txt .

Categoría	Numero del requerimiento	Descripción	Realizado
Accesibilidad	1	Fácil acceso y manipulación	✓
	2	Tener una interfaz auditiva	✓
	3	Dar las noticias a través de un sintetizador de voz	✓
	4	Recibir ordenes a través de comandos de voz (reconocedor de voz)	✓
	5	Presentar una interfaz grafica agradable para el usuario	✓
	6	Estar orientado a personas con discapacidades de visión	✓
Funcionalidad	7	Poder hacer búsquedas de múltiples paginas	✓
	8	Poder descargar de manera rápida y sin presentar perdidas de la información de las paginas con fuentes RSS	✓
	9	Ir al titular anterior	✓
	10	Pasar a la siguiente titular	✓
Configuración	11	Configuración sencilla	✓
	12	Poder agregar, modificar y eliminar paginas con contenidos RSS	✓

Tabla 4.2: Verificación de requisitos cumplidos

4.1.3. Prueba de instalación

El sistema gestor de información fue instalado exitosamente en el sistema operativo *Ubuntu 9.04 kernel 2.6.28-11-generic*, por otra parte, es indispensable la instalación de las bibliotecas *mysql++*, *curl* y *festival* en los respectivos directorios de linux para su funcionamiento.

Es importante aclarar que esta prueba se ha realizado para garantizar la instalación del gestor información al ser montado sobre un servidor ya que la aplicación ha sido diseñada para funcionar a través de una interfaz web donde solo es necesario contar con un navegador web (*mozilla firefox*, *internet explorer* o *google chrome* preferiblemente) y el link del gestor de información.

4.1.4. Prueba de recuperación

En caso de interrupción del sistema de manera forzada, indebida o accidental el gestor de información no está en capacidad de realizar ningún respaldo de la búsqueda.

Por lo tanto queda expuesto a estas fallas o interrupciones del equipo, si esta falla llegará a ocurrir el usuario deberá realizar la búsqueda de nuevo.

4.2. Certificación del componente de *software*

Luego de haber obtenido los resultados de las diferentes pruebas realizadas y haber corregido los errores para lograr la aceptación satisfactoria del producto de software, se puede concluir que durante esta fase se cumple el buen funcionamiento del gestor de información. Cumpliendo con todos los requerimientos especificados y diseño del sistema vistos en el capítulo anterior.

A continuación se procede con la certificación del sistema. La tabla 4.3 sirve para verificar todos los aspectos relacionados del sistema, comprobando cada una de sus partes y dando garantía del componente de *software* desarrollado de manera interna.

Tarea	Detalles	Validación
Verificar consistencia	El sistema es consistente en todas sus formas (especificación, diseño, pruebas, funcionamiento).	Cumple según el diseño expuesto en el capítulo 3.
Verificar dependencia	Las dependencias del gestor se han hecho correctamente, y su funcionamiento es el debido (como se ha demostrado en los casos de prueba).	Cumple con todas las dependencias.
Verificar configuración	El sistema se instala con facilidad.	Cumple, debe tenerse en cuenta las bibliotecas (mysql++, curl, festival) necesarias para la ejecución del sistema las cuales deben estar en el mismo directorio del gestor de información
Compatibilidad entre versiones	No existen versiones anteriores	Ya que es la primera versión, se cumple.
Verificar Estándares	XML y RSS para el intercambio de información	Si se cumple
Verificación de la documentación.	Las especificaciones del sistema gestor de información, las interfaces y las operaciones se exponen en el capítulo 3.	Si se cumple.
Pruebas y resultados del componente	Las pruebas realizadas en la sub-sección de este capítulo.	Si se cumple.

Tabla 4.3: Pruebas de integración de los componentes

El siguiente paso sería la certificación externa que le compete al Laboratorio *Robotics lab* de universidad Carlos III de Madrid.

4.3. Liberación del componente de *software*

La liberación del componente es la última fase del método *Watch Component*, se trata de cuando este a disposición del público el componente de *software* en un repositorio.

En este caso la liberación del componente se completará cuando se indexe en el repositorio de *Robotics lab* de la Universidad Carlos III de Madrid.

www.bdigital.ula.ve

Capítulo 5

Conclusiones y recomendaciones

5.1. Conclusiones

- El sistema gestor de información cumple con los objetivos planteados, poder realizar búsquedas en Internet y obtener resultados actualizados de sitios de interés (páginas *webs*, foros, *blogs* y redes sociales).
- A través del *software* desarrollado se logra mejorar la interacción humano-computador de personas con discapacidades visuales.
- El gestor sirve como herramienta de información de uso en línea para cualquier persona con los conocimientos básicos del manejo de un computador (personas de tercera edad a quienes en muchas ocasiones se les dificulta el manejo de un computador), también es una herramienta de accesibilidad para usuarios con discapacidades de visión.
- El sistema puede ser instalado en cualquier computador que cumpla los requisitos mínimos para su funcionamiento, a pesar que este componente de *software* ha sido diseñado como una nueva habilidad del robot Maggie.
- La reutilización de componentes adaptables al gestor de información fueron piezas clave al momento del desarrollo del producto de *software*.
- El componente de *software* Festival usado para la síntesis de voz, se logro integrar

de manera sencilla al gestor ya que este posee una biblioteca API desarrollada para C++, también este sintetizador brinda la posibilidad de ser configurado en diferentes idiomas.

- Debido a las dificultades presentadas con el componente para el reconocimiento de voz *Loquendo* (el reconocedor que posee *Maggie*, se trata de un componente de *software* privado y su licencia esta atada al *hardware* del robot, por lo que no pudo ser implantado para la demostración del gestor de información) hubo que realizar algunos ajustes para utilizar el reconocedor de voz *Dragon Naturally Speaking*, en la versión demo del gestor de información.

5.2. Recomendaciones

- El gestor de información puede ser ampliado para la reproducción de *podcast*, así como también lectura y escritura de correos electrónicos.
- Hacer del gestor de información una herramienta de información en múltiples idiomas.
- Extender el gestor de información hacia los servicios de redes sociales (*twitter*, *facebook*), entre otras. Ya que éstas cuentan con fuentes RSS y los API necesarios para su integración.
- Un campo interesante para el desarrollo de futuros trabajos de investigación es la ampliación del gestor de información, para personas con otros tipos de discapacidad, para mejorar su interacción y accesibilidad a la red.

Bibliografía

Baekdal, T. (2009), 'Where is everyone?'. <http://www.baekdal.com/articles/management/market-of-information/>, pág. consultada 27/05/09.

Barber, R. (2000), 'Desarrollo de una arquitectura para robots móviles autónomos. aplicación a un sistema de navegación topológica.', *Leganés España: Universidad Carlos III de Madrid, España* .

Breazeal, C. (2004), *Social interactions in HRI: the robot view. Man and Cybernetics*, Vol. Parte C, IEEE.

CBDi, F. (1999), 'Component based development. using componentized. cbdi forum report.'. <http://cbdiforum.org>, pág. consultada 08/07/09.

Corrales, A., Rivas, R. & Salichs, M. A. (2008), 'Habilidad de adquisición de información web para el robot personal maggie', *RoboCity 2030 : Universidad Carlos III de Madrid, España. ISBN 978-84-691-3824-3. Cap 4 pag 59-75* .

Deitel, P. (2003), *Como programar en C++*, Pearson.

Esteban, M. P. (2006), 'Sistemas avanzados de recuperación de información'. http://es.geocities.com/rss_guia_facil/que_es_rss.html, pág. consultada 07/07/09.

FOAL (2006), 'Fundación once para américa latina'. <http://foal.once.org/FOAL/es/>, pág. consultada 21/03/09.

Gorostiza, J., Barber, R., Khamis, A., Malfaz, M., Pacheco, R., R., R. & otros (2006), 'Multimodal human-robot interaction framework for a personal robot', *RO-MAN*

06: *The 15th IEEE International Symposium on Robot and Human Interactive Communication. IEEE.* .

Hammersley, B. (2005), *Developing Feeds with RSS and Atom*, O'Reilly.

Lazarum, M. d. b. d. A. (2008), 'Noticias productos y servicios de discapacidad'. <http://www.lazarum.com/>, pág. consultada 05/11/08.

Lewis, G. (1994), *What is Software Engineering*, DataPro (4015).

Mediavilla, C. (2008), 'Red y discapacidad'. <http://www.famma.org/opinion/mediavilla.htm>, pág. consultada 28/10/08.

Montilva, J. (2006), 'Desarrollo de software basado en líneas de productos de software'. <http://www.ieee.org.ar/downloads/2006-montilva-productos.pdf>, pág. consultada 28/11/08.

Nauta, A. (1995), *Informática y computación*, Ediciones Nauticas.

ONCE (2008), 'Corporación sin ánimo de lucro con la misión de mejorar la calidad de vida de las personas ciegas y con discapacidad visual de toda España'. <http://www.once.es/new/conocenos/index.html>, pág. consultada 27/10/08.

Parsons, J. J. (2008), *Conceptos de computación: Nuevas perspectivas*, Décima Edición.

R. Rivas, R. Barber, A. C. . M. A. S. (2007), 'Arquitectura de software de un robot personal.', *En Arquitecturas de Control para Robots: Universidad Politécnica de Madrid, España. pag 101 - 115* .

Ray, E. T. (2001), *Learning XML*, O'Reilly.

Robotics-Lab (2006), 'Maggie personal robots'. http://roboticslab.uc3m.es/roboticslab/robot.php?id_robot=1, pág. consultada 25/10/08.

Sanders, D. (1992), *Informática presente y futuro*, McGraw-Hill.

Violet (2007), 'Nabaztag'. http://www.violet.net/produits_us.html, pág. consultada 25/02/09.

W3C, W. A. I. (2007*a*), 'Introducción a la accesibilidad web'. <http://www.w3c.es/Traducciones/es/WAI/intro/accessibility>, pág. consultada 25/10/08.

W3C, W. A. I. (2007*b*), 'Referencias web de software desarrollado para personas con discapacidad de visión'. <http://www.w3.org/WAI/References/Browsing#1>, pág. consultada 25/10/08.

W3C, W. W. W. C. (2003), 'Extensible markup language (xml)'. <http://www.w3.org/XML/>, pág. consultada 25/10/09.

Wikipedia (2009), 'Extensible markup language'. http://es.wikipedia.org/wiki/Extensible_Markup_Language, pág. consultada 25/10/09.

www.bdigital.ula.ve

Apéndice A

Ajuste del método *Watch Component*

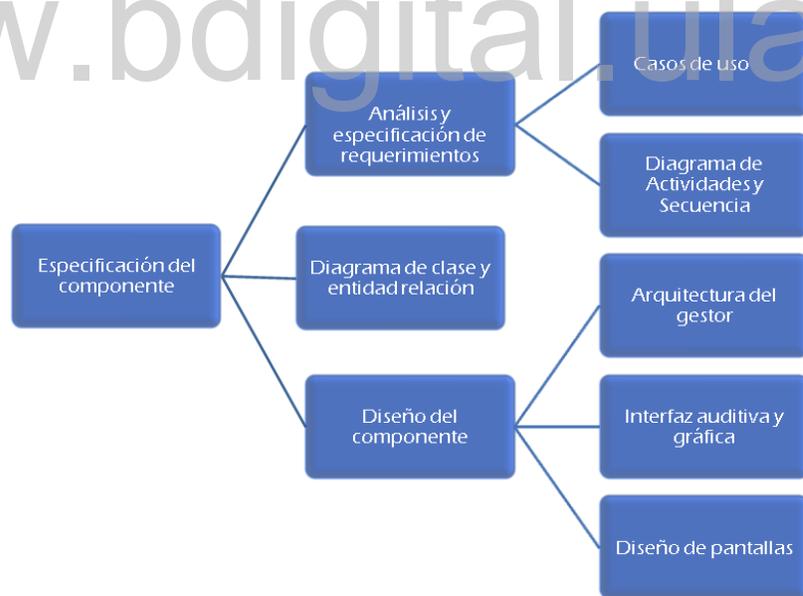


Figura A.1: Ajuste del método *Watch Component*.

Apéndice B

Tablas de especificación de requisitos

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (2)	Tipo de Requisito (F/NF): No funcional
Categoría : Accesibilidad	
Descripción: Tener interfaz auditiva	
Justificación: el gestor de información debe poseer una interfaz auditiva que le permita al usuario con discapacidad poder tener interacción humano-computador	
Fuente : desarrollador	
Prioridad del requisito (máximo) 1 2 3 (mínima): 2	
Dependencias (con otros requisitos): 3,4	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): buscar noticias	

Tabla B.1: Especificación del requisito 1

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (2)	Tipo de Requisito (F/NF): No funcional
Categoría : Accesibilidad	
Descripción: Tener interfaz auditiva	
Justificación: el gestor de información debe poseer una interfaz auditiva que le permita al usuario con discapacidad poder tener interacción humano-computador	
Fuente : desarrollador	
Prioridad del requisito (máximo) 1 2 3 (mínima): 2	
Dependencias (con otros requisitos): 3,4	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): buscar noticias	

Tabla B.2: Especificación del requisito 2

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (3)	Tipo de Requisito (F/NF): funcional
Categoría : Accesibilidad	
Descripción: dar las noticias con un sintetizador de voz	
Justificación: es necesario contar con un sintetizador de voz para transmitir las noticias, de ser posible tener opción de voz en español.	
Fuente : desarrollador	
Prioridad del requisito (máximo) 1 2 3 (mínima): 2	
Dependencias (con otros requisitos): 3	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): transmitir noticias	

Tabla B.3: Especificación del requisito 3

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (4)	Tipo de Requisito (F/NF): funcional
Categoría :Accesibilidad	
Descripción: recibir órdenes a través de comandos de voz, reconocedor de voz	
Justificación: el reconocedor debe captar palabras claves para poder utilizar el gestor de información, para este requisito se cuenta con el reconocedor implantado en el robot personal Maggie	
Fuente : desarrollador	
Prioridad del requisito (máximo) 1 2 3 (mínima): 2	
Dependencias (con otros requisitos): ninguno	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): buscar noticias y transmitir noticias	

Tabla B.4: Especificación del requisito 4

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (5)	Tipo de Requisito (F/NF): no funcional
Categoría : Accesibilidad	
Descripción: presentar una interfaz grafica agradable para el usuario	
Justificación: este requerimiento está orientado a la configuración del componente donde el usuario capacitado podrá configurar las diferentes opciones presentes.	
Fuente : desarrollador	
Prioridad del requisito (máximo) 1 2 3 (mínima): 1	
Dependencias (con otros requisitos):	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): configurar las fuentes RSS	

Tabla B.5: Especificación del requisito 5

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (6)	Tipo de Requisito (F/NF): no funcional
Categoría : Accesibilidad	
Descripción: estar orientado a personas con discapacidades de visión	
Justificación: cuenta con dos componentes (reconocedor y sintetizador de voz) encargados de brindar al usuario discapacitado un fácil acceso y manipulación del gestor de información	
Fuente : desarrollador	
Prioridad del requisito (máximo) 1 2 3 (mínima): 1	
Dependencias (con otros requisitos): 2	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): buscar noticias y transmitir noticias.	

Tabla B.6: Especificación del requisito 6

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (7)	Tipo de Requisito (F/NF): funcional
Categoría : Funcionalidad	
Descripción: Pueda hacer búsquedas de múltiples paginas	
Justificación: ambos usuarios del gestor deben poder hacer búsquedas de noticias, este es el requisito fundamental de todo el desarrollo del componente	
Fuente : usuario	
Prioridad del requisito (máximo) 1 2 3 (mínima): 1	
Dependencias (con otros requisitos): ninguno	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): buscar noticias	

Tabla B.7: Especificación del requisito 7

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (8)	Tipo de Requisito (F/NF): Funcional
Categoría : Funcionalidad	
Descripción: poder descargar de manera rápida y sin presentar pérdida de información de las fuentes RSS	
Justificación: es importante poder descargar la información y garantizar que no se pierdan datos que posteriormente deben ser utilizados por el gestor, para ello se reutilizara un componente de software que se adecue a dicho requerimiento.	
Fuente : desarrollador	
Prioridad del requisito (máximo) 1 2 3 (mínima): 1	
Dependencias (con otros requisitos): 14	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): buscar noticias	

Tabla B.8: Especificación del requisito 8

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (9)	Tipo de Requisito (F/NF): funcional
Categoría : Funcionalidad	
Descripción: Ir al titular anterior	
Justificación: si el usuario desea ir al titular anterior, el gestor de información debe estar en capacidad de responder a dicha solicitud del usuario	
Fuente : desarrollador	
Prioridad del requisito (máximo) 1 2 3 (mínima): 3	
Dependencias (con otros requisitos): 7	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): transmitir noticias	

Tabla B.9: Especificación del requisito 9

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (10)	Tipo de Requisito (F/NF): funcional
Categoría : Funcionalidad	
Descripción: pasar al siguiente titular	
Justificación: el usuario puede solicitar pasar al siguiente titular de manera inmediata	
Fuente : desarrollador	
Prioridad del requisito (máximo) 1 2 3 (mínima): 2	
Dependencias (con otros requisitos): 7	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): transmitir noticias	

Tabla B.10: Especificación del requisito 10

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (11)	Tipo de Requisito (F/NF): funcional
Categoría : Configuración	
Descripción: configuración sencilla	
Justificación: en este caso el usuario invidente debe contar con una persona capacitada para poder realizar la configuración del gestor de información	
Fuente : usuario	
Prioridad del requisito (máximo) 1 2 3 (mínima): 1	
Dependencias (con otros requisitos): 1	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): configurar las fuentes RSS	

Tabla B.11: Especificación del requisito 11

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (12)	Tipo de Requisito (F/NF): funcional
Categoría : Configuración	
Descripción: Poder agregar, modificar y eliminar paginas con contenidos RSS	
Justificación: el usuario capacitado tiene un conjunto de opciones que le permite realizar el sistema para la configuración de las fuentes URL con contenidos RSS, él puede agregar, modificar o eliminar las paginas webs en la base de datos del gestor de información	
Fuente : usuario	
Prioridad del requisito (máximo) 1 2 3 (mínima): 1	
Dependencias (con otros requisitos): ninguno	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): configurar las fuentes RSS	

Tabla B.12: Especificación del requisito 12

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (13)	Tipo de Requisito (F/NF): No funcional
Categoría : Hardware	
Descripción: ser una nueva habilidad para el robot personal Maggie	
Justificación: el gestor de información es un nuevo componente de software para el robot personal Maggie	
Fuente : desarrollador	
Prioridad del requisito (máximo) 1 2 3 (mínima): 1	
Dependencias (con otros requisitos): ninguno	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): ninguno	

Tabla B.13: Especificación del requisito 13

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (14)	Tipo de Requisito (F/NF): no funcional
Categoría : Hardware	
Descripción: tener acceso a internet	
Justificación: es indispensable contar con acceso a internet ya que el componente tiene una interfaz web	
Fuente : desarrollador	
Prioridad del requisito (máximo) 1 2 3 (mínima): 1	
Dependencias (con otros requisitos): ninguno	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): ninguno	

Tabla B.14: Especificación del requisito 14

www.bdigital.ula.ve

Ficha: recolección de requisitos del gestor de información	
Número de Requisito (15)	Tipo de Requisito (F/NF): no funcional
Categoría : Hardware	
Descripción: tener periféricos multimedia (micrófono y parlantes)	
Justificación: es necesario que el computador cuente con dichos periféricos, ya que el gestor cuenta con un sintetizador y reconocedor de voz que sirven para la interacción humano computador	
Fuente : desarrollador	
Prioridad del requisito (máximo) 1 2 3 (mínima): 1	
Dependencias (con otros requisitos): ninguno	
Conflictos (con otros requisitos): ninguno	
Casos de uso que lo necesitan (identificación): ninguna	

Tabla B.15: Especificación del requisito 15

Apéndice C

Diagramas de secuencia

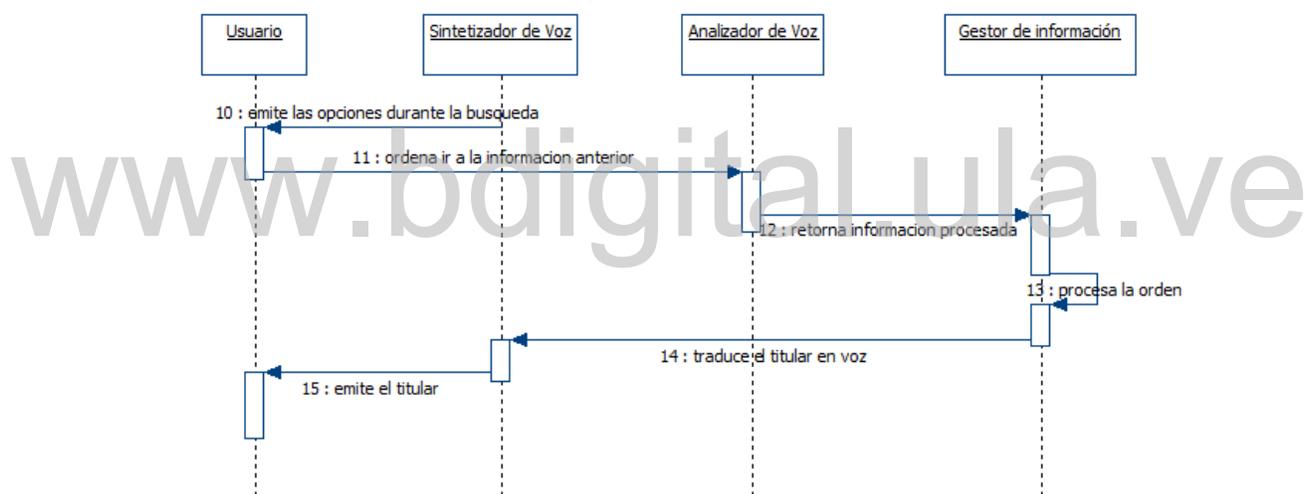


Figura C.1: Diagrama de secuencia noticia anterior.

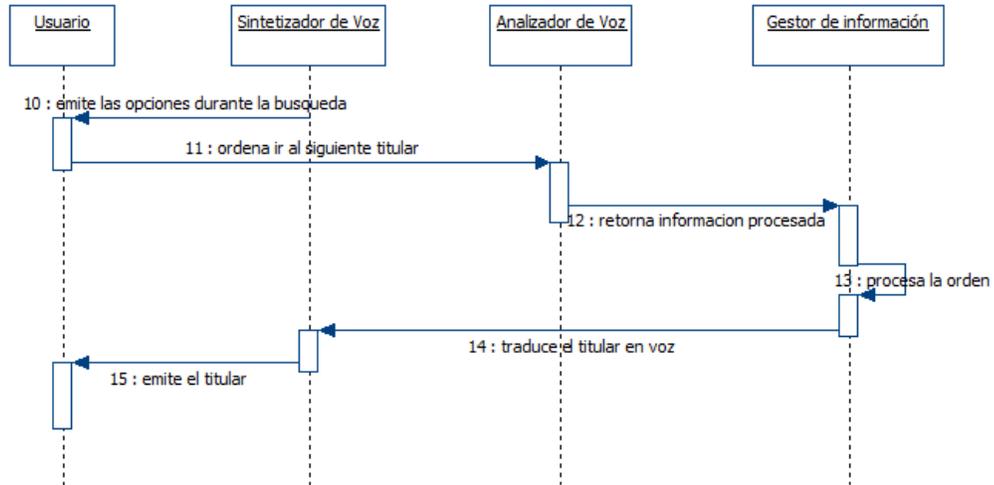


Figura C.2: Diagrama de secuencia noticia siguiente.

www.bdigital.ula.ve

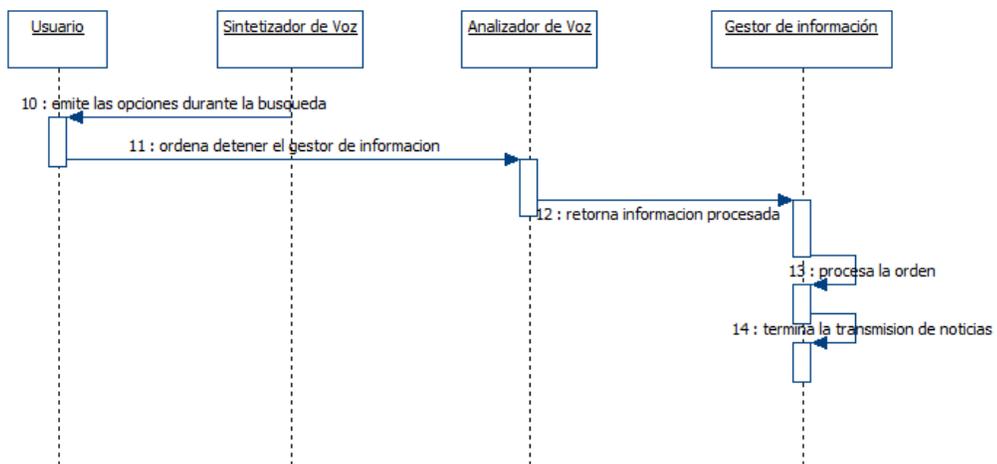


Figura C.3: Diagrama de secuencia detener el gestor de información.

Apéndice D

Diagramas de clase implementado y tablas de especificación de las clases

Nombre	Descripción	Pre condición	Post condición
añadir_url()	Permite añadir una nueva dirección URL a la base de datos de fuentes RSS	Ninguna	Nueva URL añadida
modificar_url()	Modificar la información de alguna de las fuentes de RSS de la BD	Debe existir la fuente a modificar	Estado de la modificación exitosa o no exitosa
eliminar_url()	Permite eliminar una de las fuentes RSS de la BD	Debe existir la fuente a eliminar	Estado de la eliminación exitoso o no exitoso
listar_url()	Lista todas las fuentes de RSS almacenadas en la BD	Debe existir al menos una fuente de RSS en la BD	Retorna la lista de fuentes RSS

Tabla D.1: Especificación de la clase configuración.

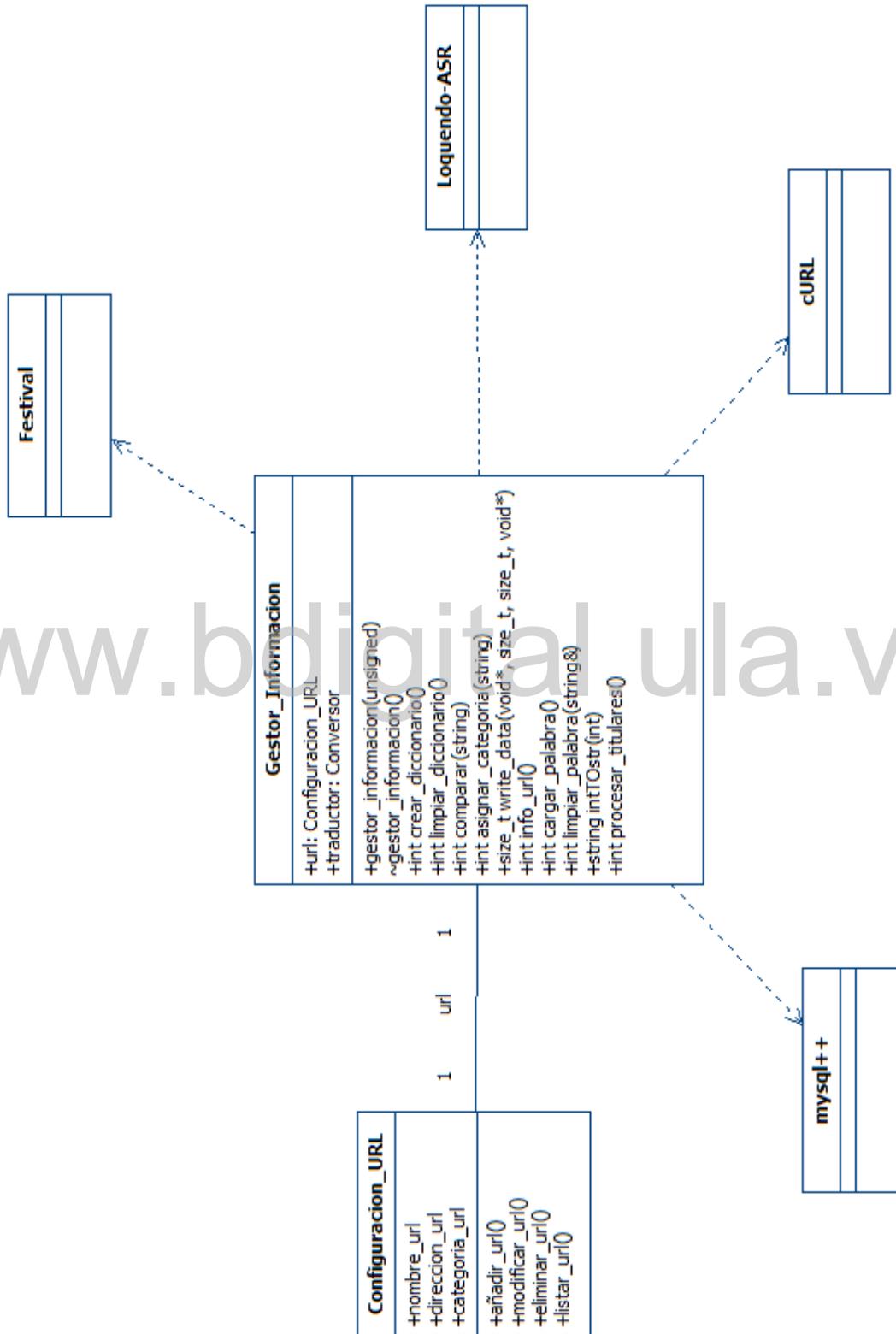


Figura D.1: Diagrama de clases.

Nombre	Descripción	Pre condición	Post condición
gestor_informacion(unsigned)	Constructor gestor de información	Ninguna	Inicio de la aplicación
~gestor_informacion()	Destructor gestor de información	Haber iniciado la aplicación	Detener aplicación
int crear_diccionario()	Crea el diccionario a partir de las categorías existentes en la base de datos	Existen categorías en la base de datos	Diccionario creado
int limpiar_diccionario()	Elimina palabras repetidas en el diccionario	Exista el diccionario	Diccionario sin palabras repetidas
int comparar(string)	Función encargada de comparar lo que el usuario desea buscar con el diccionario interno de la aplicación.	La cadena string tenga debe tener algún valor asignado	Palabra reconocida
int asignar_categoria(string)	Asigna la búsqueda de la categoría correspondiente en BD de fuentes de RSS	Haber reconocido la cadena string correspondiente a la búsqueda del usuario	Devuelve un entero indicando si se asigna la categoría.
size_t write_data(void*, size_t, size_t, void*)	Se encarga de guardar la información descargada de internet en un archivo XML	Haber realizado la descarga de la fuente de información a través de internet	Documento XML correspondiente a la descarga
int info_url()	Descarga la información de la fuente a través de Internet	Tener la URL correspondiente para la descarga	Información de la URL descargada.
int cargar_palabra()	Función encargada de verificar las palabras válidas	Tener el documento XML	Devuelve un entero si se encontró
int limpiar_palabra(string&)	Esta función limpia las palabras de acentos y símbolos extraños del documento XML	Tener el documento XML	Crea un nuevo archivo con el documento limpio.
int intToStr(int)	Convierte de entero a cadena	Ninguna	Conversión de int a string
int procesar_titulares()	Procesa el documento XML eliminando todo el código XML innecesario para la transmisión de las noticias	Tener el documento XML	Indica si el proceso se cumplió con éxito y guarda el archivo definitivo para la transmisión del titular

Tabla D.2: Especificación de la clase gestor información.

Apéndice E

Capturas de pantallas de la interfaz grafica de configuración



The screenshot shows a web application interface with a menu bar at the top containing the options: Insertar, Modificar, Eliminar, Consultar, and Listar. Below the menu bar, the main content area is titled "Datos de la URL". This area contains a form with three input fields: "Nombre:", "Dirección:", and "Categoría:". At the bottom of the form, there are two buttons: "Borrar" and "Insertar Datos". A large, semi-transparent watermark "www.bdigital.ula.ve" is overlaid across the center of the screenshot.

Figura E.1: Captura de pantalla, insertar.



The screenshot shows a web interface with a menu bar at the top containing the following options: Insertar, Modificar, Eliminar, Consultar, and Listar. The 'Modificar' option is highlighted. Below the menu bar, the main content area is titled 'Indique el Nombre de la URL a Modificar'. Underneath this title, there is a text input field labeled 'Nombre' and a button labeled 'Submit Query'.

Figura E.2: Captura de pantalla, modificar.

www.bdigital.ula.ve



The screenshot shows a web interface with a menu bar at the top containing the following options: Insertar, Modificar, Eliminar, Consultar, and Listar. The 'Eliminar' option is highlighted. Below the menu bar, the main content area is titled 'Seleccione la URL a Eliminar'. Underneath this title, there is a dropdown menu labeled 'Nombre de la URL' and a button labeled 'Seleccionar'.

Figura E.3: Captura de pantalla, eliminar.

Inserir Modificar Eliminar Consultar Listar

Listado de los URL

ID	Nombre	Direccion	Categoria
7	El mundo futbol	http://rss.elmundo.es/rss/descarga.htm?data2=59	futbol
5	AS DEPORTES	HTTP://WWW.AS.COM/RSS/FEED.HTML?FEEDID=61	futbol
8	Globovision	http://globovision.com/rssbychannel.php?l=es&cha=4	deportes
9	BBC PORTADA	http://www.bbc.co.uk/mundo/index.xml	portadas
10	El mundo PORTADA	http://rss.elmundo.es/rss/descarga.htm?data2=4	portadas

www.bdigital.ula.ve

Figura E.4: Captura de pantalla, listar.