

# UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

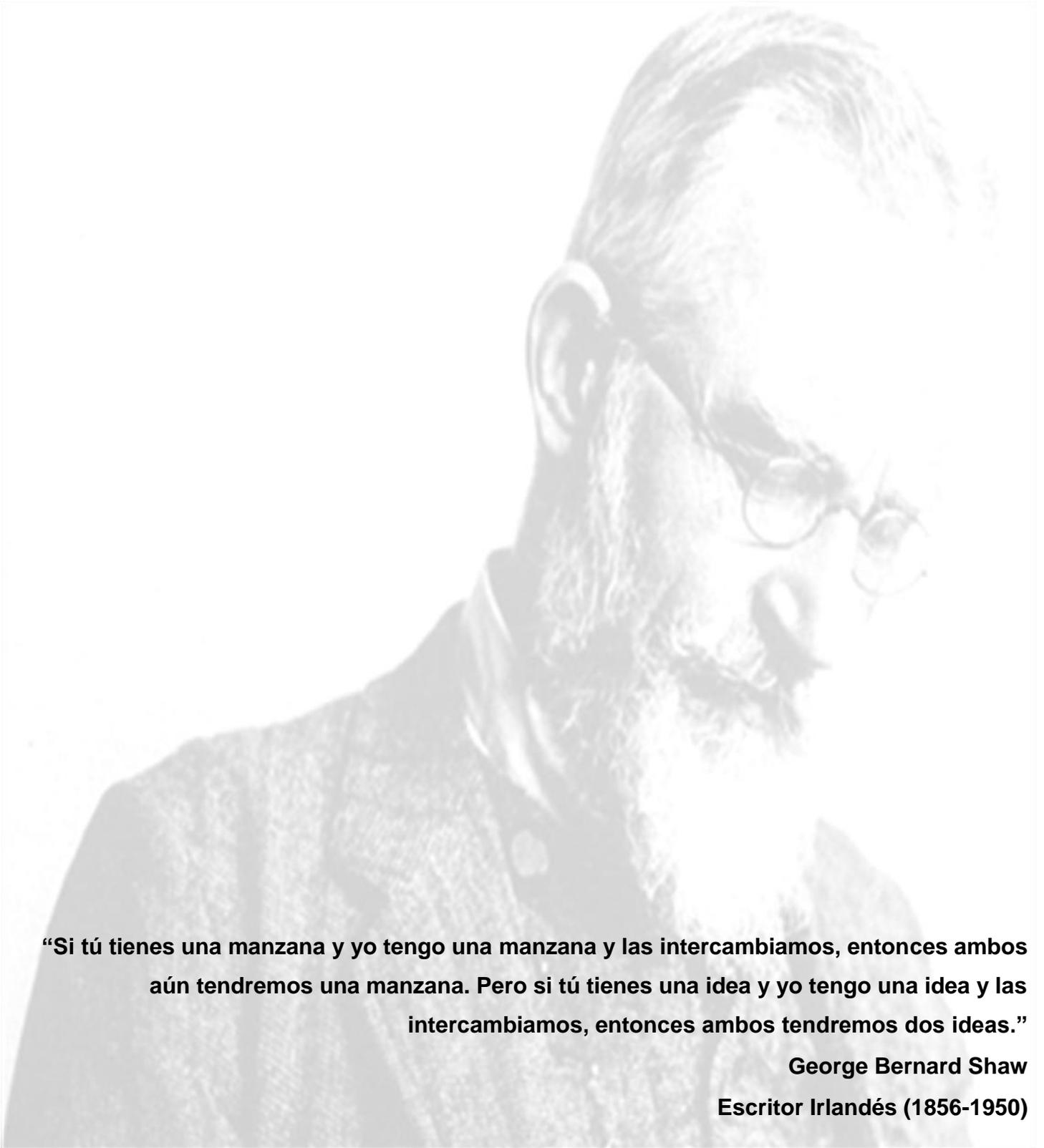
## FACULTAD 4



**TÍTULO:** Componente para la integración de un lector de pantalla con el marco de trabajo Xalix.

**AUTORA:** Yuliexa Batista Morell

**TUTOR:** Ing. Adrián García Sánchez



**“Si tú tienes una manzana y yo tengo una manzana y las intercambiamos, entonces ambos aún tendremos una manzana. Pero si tú tienes una idea y yo tengo una idea y las intercambiamos, entonces ambos tendremos dos ideas.”**

**George Bernard Shaw  
Escritor Irlandés (1856-1950)**

## Declaración de autoría

Declaro que soy la única autora del trabajo “Componente para la integración de un lector de pantalla con el marco de trabajo Xalix” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Autor: Yuliexa Batista Morell Tutor: Ing. Adrián García Sánchez**

## Agradecimientos

*A mi mamá y a mi papá por darme la vida y guiarme siempre por el buen camino, predicando con el ejemplo. Gracias por creer en mí, por sus sacrificios y por todos sus esfuerzos.*

*A mi hermana Yunieska por todo su apoyo y preocupación, aún desde la lejanía, por quererme y estar siempre para mí, sin importar cuán difícil pudiera ser.*

*A mi tío Tinito por ser tan cercano a mí, quererme y ayudarme siempre que lo necesito.*

*A mi tía Mimi, que aunque ya no está físicamente presente, sus consejos siguen impregnados en mis días. En general a toda mi familia, que de una forma u otra han sido partícipes de este inmenso sueño.*

*A mi novio Roberto porque me ayudó y apoyó en todo. Porque durante todo este tiempo, ha sido mi amigo, mi compañero y mi guía. Por darme las fuerzas para continuar y brindarme todo su amor y cariño.*

*A mi tutor Adrián por ayudarme y guiarme durante toda la investigación y durante el tiempo que he estado en el proyecto Alfaomega.*

*A mis amigos y amigas por su apoyo, su cariño y su respeto; sepan que estarán siempre conmigo.*

*A mis profesores porque de ellos aprendí todo lo que se.*

*A la Revolución y en especial a Fidel por darme la oportunidad de estudiar en la UCI.*

*En fin doy las gracias a todas las personas que estuvieron a mi lado a lo largo de este sueño, que hoy se está haciendo realidad.*

## Dedicatoria

*A mi familia en general por lo que significan para mí y por darme todo su apoyo en cada momento.*

*A mi mamá Juana que es la persona más importante en mi vida y una de las que más quiero.*

*A mi papá Idelfonso, a mi hermana Yuneska y a mi tío Celestino, por ser tan buenos conmigo, por estar siempre ahí cuando más los necesito, por darme tanto amor y cariño, por lograr hacer de mí una persona de bien y una profesional.*

*A mi novio Roberto que ha sido mi guía sobre todo profesionalmente y ha estado a mi lado cuando más lo he necesitado.*

*Muy especialmente también a mi tía Deysi, que aunque no estuvo presente en esta etapa de mi vida, la quiero, la respeto y siempre la tengo presente.*

*A todos ustedes va dedicado este trabajo. Gracias.*

*Yuliexa Batista Morell*

## Resumen

El ser humano desde su surgimiento ha sentido la necesidad de crear un continuo proceso de mejora y simplificación de sus condiciones de vida; mediante la imitación artificial de sus características naturales. Entre estas, es posible destacar el habla; característica que las TIC han facilitado su imitación, mediante el uso de lectores de pantalla y sintetizadores de voz.

Los lectores de pantalla son de gran utilidad en momentos en que los ojos deben estar enfocados en otras actividades de mayor prioridad o para personas con deficiencias visuales, que les impida acceder a los programas de una computadora, principalmente a sitios web. Por lo que el presente trabajo estuvo encaminado a desarrollar un componente que funcionara como lectores de pantalla en la web y además permita integrar sintetizadores de texto compatibles con cualquier sistema operativo.

Para cumplir el objetivo trazado se realizó un análisis profundo de las tecnologías y herramientas existentes para el desarrollo de software, seleccionando las más adecuadas, así como también se llevaron a cabo todos los pasos necesarios que propone la metodología empleada (RUP). Obteniendo como resultado final una solución que cumple con todo lo definido durante la investigación y que contiene la calidad requerida.

**Palabras claves:** lector de pantalla, xalix, symfony 2, bundle, sintetizador de voz.

# Índice

Introducción .....	1
Capítulo 1: Fundamentación teórica .....	6
1.1 Introducción.....	6
1.2 Investigación teórica.....	6
1.3 Componentes de un lector de pantalla .....	7
1.4 Sintetizadores de voz y su evolución.....	7
1.5 Procesos para la síntesis de voz .....	8
1.6 Normalización de texto .....	9
1.7 Tecnologías para generar ondas de voz sintética .....	11
1.8 Análisis de las soluciones similares.....	12
1.9 Tendencias y tecnologías.....	13
1.10 Herramientas de síntesis de voz .....	13
1.11 Sintetizadores de voz seleccionados.....	15
1.12 Metodologías de desarrollo de software .....	15
1.12.1 Metodologías tradicionales .....	16
Proceso Unificado de <i>Rational</i> ( <i>RUP</i> en lo adelante).....	16
1.12.2 Metodologías ágiles .....	16
Programación Extrema ( <i>XP</i> en lo adelante) .....	17
1.13 Metodología seleccionada.....	17
1.14 Herramientas CASE (Ingeniería de software asistida por computadoras, en español) .....	18
1.15 Herramientas CASE para el modelado UML .....	19
1.16 Selección de una herramienta CASE para el modelado UML.....	21

1.17	Selección del lenguaje a utilizar .....	22
1.18	Framework para el desarrollo.....	22
	Capa lógica del negocio .....	22
	Capa de acceso a datos.....	23
1.19	Sistema gestor de base de datos .....	23
1.20	Selección de un gestor de base de datos:.....	24
1.21	Entorno de desarrollo integrado ( <i>IDE</i> en lo adelante) .....	24
1.22	Selección de un IDE.....	25
1.23	Servidor web .....	25
1.24	Selección del servidor Web a utilizar .....	26
1.25	Conclusiones.....	26
Capítulo 2: Características del componente.....		27
2.1	Introducción.....	27
2.2	Modelo del dominio .....	27
	2.2.1    Conceptos del dominio.....	27
	2.2.2    Modelo del dominio .....	28
2.3	Descripción de las funcionalidades propuestas.....	28
2.4	Requisitos .....	29
	2.4.1    Requisitos funcionales .....	29
	2.4.2    Requisitos no funcionales .....	31
2.5	Actores y casos de uso del sistema .....	32
	2.5.1    Descripción de los actores del sistema.....	32
	Tabla 1. Actores del sistema .....	32
	2.5.2    Diagrama de casos de uso del sistema.....	32

**ÍNDICE | UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS | FACULTAD 4**  
**COMPONENTE PARA LA INTEGRACIÓN DE UN LECTOR DE PANTALLA AL MARCO DE TRABAJO XALIX**

2.5.3	Descripción de los casos de uso del sistema .....	34
2.6	Conclusiones.....	39
Capítulo 3: Análisis y diseño del componente .....		40
3.1	Introducción.....	40
3.2	Modelo de análisis .....	40
3.2.1	Diagramas de clases del análisis .....	41
3.2.2	Diagramas de colaboración.....	42
3.3	Patrón arquitectónico Modelo Vista Controlador (en lo adelante MVC) en Symfony .....	42
3.4	Patrones de diseño .....	43
Patrones GRASP .....		44
Inyección de Dependencias (DI en lo adelante por sus siglas en inglés) .....		45
Patrones GoF .....		45
3.5	Modelo del diseño .....	45
3.6	Diagrama de clases del diseño.....	45
3.7	Diagrama de Despliegue .....	47
3.8	Diseño de la base de datos .....	48
3.9	Descripción de las tablas de la base de datos.....	48
3.10	Conclusiones.....	50
Capítulo # 4 Implementación y prueba de las funcionalidades del componente .....		51
4.1	Introducción .....	51
4.2	Modelo de implementación .....	51
4.2.1	Diagramas de componentes.....	51
4.3	Pruebas de software .....	53
4.3.1	Niveles de pruebas.....	53

4.3.2 Métodos de prueba.....	55
4.3.3 Diseño de Casos de Pruebas.....	57
4.3.4 Resultados obtenidos.....	60
4.4 Conclusiones.....	62
Conclusiones.....	63
Recomendaciones.....	64
Referencias bibliográficas.....	65
Anexos.....	69
Anexo # 1 Instalación del componente.....	69
Anexo # 2 Implementación e integración de nuevos sintetizadores de texto a voz al componente.....	71
Anexo # 3 Etiquetas áreas para el funcionamiento del lector de pantalla.....	75
Anexo # 4 Descripciones de los casos de uso.....	80
Anexo # 5: Diagramas de clases del análisis.....	92
Anexo # 6 Diagramas de colaboración.....	93
Anexo # 7 Diagramas de diseño de clases de los casos de uso.....	95
Anexo # 8 Diagramas de componentes.....	98
Anexo # 9 Descripción de los casos de pruebas.....	100

# Introducción

El ser humano desde su surgimiento ha sentido la necesidad de tener un continuo proceso de mejora y simplificación de sus condiciones de vida, mediante la imitación artificial de sus características naturales. Las tecnologías de la información y comunicación (**TIC**<sup>1</sup> en lo adelante) hacen que este proceso sea más eficiente, favoreciendo la conversión de texto a voz y su aplicación en lectores de pantallas, muy útil en las situaciones en las cuales los ojos deben estar enfocados en otras tareas y para personas con discapacidades, que les impidan leer correctamente lo que se muestra a través de los monitores de las computadoras.

Según (Luján, 2007) los **lectores de pantalla** son herramientas asistidas que intentan identificar e interpretar lo que se muestra en la pantalla de un ordenador, con el objetivo de representar lo que identifique o interprete de forma alternativa, normalmente mediante voz, para permitir a las personas que presenten discapacidades visuales, emplear un ordenador. Los **sintetizadores de voz** son el núcleo fundamental de los lectores de pantalla, ya que son la herramienta que convierte el texto en audio. (WebAcademia, 2013)

Los lectores de pantallas tienen características; que varían de acuerdo a sus productores y las necesidades básicas u objetivas que persigan satisfacer. Algunas como *BrowseAloud*, *NVDA*, *MexVox*, y *JAWS* están disponibles solo en el sistema operativo Windows y tienen soporte para distintos idiomas; otros como *CLiCk*, *Speak y Fire Vox*, *WebAnywhere*, *SonoWeb* y *VozMe* hacen uso del *plugin* de audio que tiene el navegador Firefox Mozilla para de una forma u otra traducir el contenido de los sitios web. Orca cuenta con un magnificador de pantalla para el sistema de escritorio GNOME y *Window-Eyes* compatible con los navegadores Internet Explorer y Mozilla Firefox. Los casos anteriores son una pequeña muestra de la diversidad de software que se dedican a la interpretación, generalmente mediante el habla de lo que se muestra en la pantalla de un ordenador, pero han demostrado tener las siguientes deficiencias:

---

<sup>1</sup> Las TIC son conocidas por ser potentes instrumentos para la compensación de desigualdades. (Rodilla, 2013)

- ✓ **los lectores de pantallas que no están implementados para ser utilizados en la web no son multiplataforma.**
- ✓ **los lectores de pantallas que su utilidad está basada en la web funcionan con muy pocos navegadores.**
- ✓ **En general no permiten seleccionar de las páginas web solo el texto que se desea escuchar, sino se generan archivos completos de las páginas para ser reproducidos.**

En Cuba se han tenido grandes avances en la tecnología, sobre todo en materia de informática; pero aún queda mucho camino por recorrer y experimentar, en lo relacionado a su estudio y aplicación. La utilización de los lectores de pantalla es un ejemplo de un campo que no ha sido explotado a cabalidad.

FORTES, es un centro de desarrollo, que pertenece a la Universidad de las Ciencias Informáticas (UCI). Actualmente se encuentra trabajando en la implementación y aplicación del marco de trabajo Xalix, el cual no es más que un marco tecnológico, basado en el framework de desarrollo Symfony 2, que se rige por pautas y permite la integración de componentes de forma organizada, que puedan ser reutilizadas en favor de satisfacer las necesidades del cliente para el cual se esté trabajando. (Broche, 2014) (Díaz, et al., 2013) Actualmente los sitios web que se apoyan en el uso de esta tecnología, no cuentan con una estructura de clases flexible y extensible para la inclusión de lectores de pantallas.

El contexto anterior define una **situación problemática** que genera el siguiente **problema de investigación**: ¿Cómo integrar un lector de pantallas al marco de trabajo Xalix, para facilitar la accesibilidad a la web a personas con deficiencias visuales?

Después de haber planteado la dificultad identificada a solucionar en la investigación, se delimita como **objeto de estudio**: los lectores de pantalla.

El **campo de acción** hace referencia a las técnicas para la integración de lectores de pantallas a software de terceros.

Se plantea como **idea a defender** que si se desarrolla un complemento que integre un lector de pantalla al marco de trabajo Xalix, este contribuirá a que todos los usuarios tengan acceso a sitios web desarrollado con esa tecnología, sin importar que tengan deficiencias visuales.

El **objetivo general** que se persigue lograr consiste en desarrollar un componente para la integración de un lector de pantalla al marco de trabajo Xalix.

Para dar cumplimiento al objetivo general propuesto se definen los siguientes **objetivos específicos**:

- ✓ Identificar los procesos que intervienen en el funcionamiento de los lectores de pantalla.
- ✓ Identificar los componentes de cada proceso.
- ✓ Realizar el estudio del estado del arte sobre los componentes identificados atendiendo a su portabilidad, flexibilidad y capacidad de integrarse con terceros software.
- ✓ Definir los medios de comunicación de los componentes con terceros software.

**Posibles resultados:**

Luego de realizar una profunda investigación sobre los lectores de pantalla se espera elaborar un componente que integre un lector de pantallas al marco de trabajo Xalix.

**Métodos Teóricos** utilizados en la investigación:

- ✓ **Analítico-sintético** para el análisis detallado de distintos documentos sobre sintetizadores de voz y lectores de pantallas, permitiendo arribar a conclusiones estratégicamente apropiadas.
- ✓ **Inductivo-deductivo** para estudiar los sintetizadores de voz y lectores de pantallas, logrando arribar a conclusiones generales sobre las técnicas de integración entre ambas herramientas con la web.
- ✓ **Histórico-lógico** para conocer la evolución de las técnicas de integración de lectores de pantallas en la web.

**Métodos Empíricos** utilizados en la investigación:

- ✓ **Observación** para recopilar información sobre lectores de pantallas, sus componentes principales y las técnicas existentes para integrar estos en la web.

- ✓ **Modelación**, para realizar los casos de uso y los diferentes diagramas.
- ✓ **Análisis de fuentes de información** para examinar las distintas fuentes de información como materiales en Internet o impresos (libros y revistas).

El contenido del trabajo presenta la siguiente Estructura Capitular:

**Capítulo 1:** Fundamentación teórica.

Este capítulo presenta una descripción del estado del arte sobre cómo integrar lectores de pantallas al marco de trabajo Xalix, para facilitar la accesibilidad a la web a personas con deficiencias visuales; y las soluciones similares existentes. Se enuncian las herramientas, lenguajes y metodologías que se ajustan al desarrollo del trabajo, fundamentando su selección. Además se brinda información teórica que sustenta toda la investigación.

**Capítulo 2:** Características del componente.

En este capítulo se realiza una descripción de las características del sistema. Se lleva a cabo el levantamiento de requisitos, del cual se obtiene el modelo de dominio, los diagramas de casos de uso del sistema y sus relaciones.

**Capítulo 3:** Análisis y diseño del componente.

En este capítulo se realiza el análisis y diseño del sistema. Se obtiene el modelo de datos y la descripción de cada una de las tablas que darán solución al problema planteado. Se modelan los diagramas de clases de análisis, diagramas de iteración, diagrama entidad relación de la base de datos, se describen todas las tablas y se realiza el diagrama de despliegue definido correctamente; todo para dar paso a la implementación.

**Capítulo 4:** Implementación y prueba.

En este capítulo se muestran los artefactos generados durante la fase de implementación del componente que dará solución al problema presentado, haciendo uso de los lenguajes de programación, metodologías y herramientas seleccionados con anterioridad. Además se realizan los diagramas de componentes y se

describen los casos de prueba para cada uno de los casos de uso. Se realizan las pruebas y se documenta el proceso hasta la obtención de los resultados finales para demostrar la calidad del sistema.

# Capítulo 1: Fundamentación teórica

## 1.1 Introducción

En el presente capítulo, se realiza un estudio teórico sobre lectores de pantalla y sus características fundamentales. Se seleccionan las herramientas necesarias para la implementación de un componente que permita integrarlos con el marco de trabajo Xalix. Se estudian las soluciones existentes con el fin de encontrar una guía que sirva de base para el desarrollo del trabajo de diploma. Se analizan las metodologías de desarrollo de software y se selecciona la más apropiada.

## 1.2 Investigación teórica

Las TIC, son herramientas computacionales e informáticas que procesan, sintetizan y recuperan información representada de las más variadas forma. Además permiten el acceso a la información, para dar forma, registrar, almacenar y difundir contenidos digitalizados extraídos de diversas monografías y ensayos. Las TIC son instrumentos y materiales de construcción que facilitan el aprendizaje de distintas formas y el desarrollo de habilidades. Permiten diferenciar estilos y ritmos de los aprendices y aportan mejoras en la comunicación, la información y el desarrollo cognitivo. (Rodilla, 2013) Las TIC permiten potenciar habilidades con el fin de satisfacer las necesidades de los seres humanos, sobre todo son una fuente de recursos capaz de lograr minimizar las diferencias causadas por la presencia de alguna discapacidad física. Existe un conjunto de herramientas y software que conforman las TIC, entre ellas los lectores de pantallas.

El correcto funcionamiento de los lectores de pantalla requiere de los siguientes procesos:

- ✓ Definir el texto a reproducir.
- ✓ Normalizar el texto definido.
- ✓ Convertir el texto en un archivo de audio, mediante la utilización de un sintetizador de voz.

- ✓ Reproducir el archivo de audio.

### 1.3 Componentes de un lector de pantalla

Los lectores de pantalla para realizar el trabajo de interpretación de texto requieren de tecnologías y lenguajes de programación, como JavaScript y WAI-ARIA<sup>2</sup> que permitan la **selección** de texto. Además se necesita un **sintetizador de voz** para la normalización y conversión de texto en audio.

### 1.4 Sintetizadores de voz y su evolución

Según (Olabe, 2013) un sintetizador de voz consiste en cambiar los símbolos gráficos de un texto en lenguaje hablado. En artículos publicados por (WebAcademia, 2013), se encontró una definición más completa de lo que es un sintetizador de voz, que consiste en la producción artificial de habla humana, mediante un sistema informático que puede ser implementado en software o hardware de los productos; un texto de sistema de voz convierte el texto normal del lenguaje en el habla y otros sistemas hacen representaciones lingüísticas simbólicas como transcripciones fonéticas en el habla.

Los primeros intentos de producción artificial de voz humana, se realizaron mediante dispositivos mecánicos. El siguiente paso consistió en la construcción de dispositivos eléctricos, para llegar en los últimos años a sistemas desarrollados gracias al creciente avance de la informática. (Fernández, 1997) Los primeros sintetizadores que surgieron eran vistos como algo puramente experimental y elitista, sus orígenes datan de 1779, cuando el científico danés Cristiano Kratzenstein, trabajando en la Academia de Ciencias de Rusia, construyó modelos del tracto vocal humano que podría producir las cinco vocales largas. Luego la "máquina acústica de voz-mecánico" fue accionada por Wolfgang von Kempelen de Presburgo, Hungría, que se describe en un documento de 1791, añadiendo modelos de la lengua y los labios, que permiten producir consonantes, así como las vocales. Más tarde en 1837, Charles Wheatstone

---

<sup>2</sup> WAI -ARIA es una especificación técnica que proporciona un marco para mejorar la accesibilidad e interoperabilidad de los contenidos y aplicaciones web. Está diseñado para proporcionar información semántica sobre los objetos cuando los lenguajes principales carecen de una semántica nativa para el objeto.

produjo una "máquina parlante", basada en el diseño de Von Kempelen, por M. Faber quien en 1857 construyó la "*Euphonia*". (WebAcademia, 2013)

Los intentos anteriores de recrear la voz humana, dieron paso al sintetizador de voz más antiguo que se conoce; el **Telharmonium**, que data de 1900 y su creador fue Thaddeus Cahill's. Esto dio paso a nuevos intentos de generar los sonidos, pero no fue hasta 1920, que el Dr. Carl Emil inventara el **audiómetro**, que preconiza un futuro de nuevos sonidos electrónicos. En la década de 1930, Bell Labs desarrollaron el *vocoder*, que analiza automáticamente la voz en su tono y resonancia fundamental. A partir de su trabajo en el *vocoder*, Homer Dudley desarrolló un sintetizador de voz del teclado que funciona llamado *The Voder*, que expuso en la feria 1939 de mundo de Nueva York. (WebAcademia, 2013)

Los primeros sistemas de síntesis de voz basados en computadoras fueron creadas a finales de 1950 (WebAcademia, 2013) y en 1955 bajo el patrocinio de la RCA se construyen los sintetizadores MARK I y MARK II. En 1961, el físico John Larry Kelly, Jr y su compañero Louis Gerstman utilizaron una computadora IBM 704 para sintetizar discurso, un evento entre los más destacados en la historia de los Laboratorios Bell y en 1964 Robert Moog junto con un grupo de colaboradores creó el *Minimoog*, con el objetivo de sintetizar sonidos y crear con ellos música mediante un teclado. El primer texto en inglés, de sistema de voz fue desarrollado por Noriko Umeda, en 1968 en el Laboratorio de Electrotécnica, Japón. Otros dispositivos se produjeron principalmente con fines educativos, como el *Speak y Spell*, producido por Texas Instruments en 1978. A comienzos de los '80 se desarrolló también el *sampler* o muestreador, que permitía grabar sonidos reales y reproducirlos. (Oviedo, 2013)

### **1.5 Procesos para la síntesis de voz**

Los sintetizadores de voz que han surgido a lo largo de la historia, han diferido en tamaño de unidades de habla almacenadas y en uso de los dominios específicos, del almacenamiento de las palabras o frases enteras, que permiten la salida de alta calidad. Los sintetizadores de voz pueden incorporar un modelo del tracto vocal y otras características de la voz humana para crear salidas de voz sintética. Todos se componen de dos procesos fundamentales *front-end* y *back-end* que se encargan de proporcionar naturalidad e inteligibilidad al sonido de las voces. (WebAcademia, 2013)

El *front-end* tiene dos tareas principales; **convertir el texto** en bruto en el equivalente de la palabra escrita-*out*, o proceso conocido como normalización de texto, pre-procesamiento o *tokenización* y mediante la conversión de texto a fonema o grafema a fonema, **conformar la representación lingüística simbólica** que se emite, mediante informaciones prosodia y transcripciones fonéticas. Por otra parte, el *back-end*, se encarga de **convertir la representación lingüística simbólica** en sonido. En algunos sistemas, esta parte incluye el cálculo de la prosodia de destino, que se impone en la voz de salida. (WebAcademia, 2013)

### 1.6 Normalización de texto

Según (Arnet, 2014) la **normalización** se basa en la necesidad de encontrar una representación del conjunto de relaciones que en el proceso de actualización sea más adecuada. La licenciada en Ciencias de la Computación (Pachersky, 2012) define el **pre-procesamiento o normalización del texto de entrada** como el proceso que convierte el texto en palabras pronunciables como serían las leídas por un hablante humano. Basándose en las definiciones anteriores se decide concluir que la normalización de texto surge con la necesidad de encontrar relaciones adecuadas en la fonética, para representar de manera sencilla y entendible los textos que se muestran en pantalla. Existen tres fases fundamentales en la normalización de textos, ellas son:

- ✓ Segmentación/*tokenización* de las palabras.
- ✓ Normalización del formato de las palabras.
- ✓ Segmentación de las oraciones en el texto.

La normalización de cadenas de texto es uno de los temas recurrentes en el desarrollo de procesos ETL<sup>3</sup>. La forma más sencilla de realizar esta actividad es identificar las palabras separando los caracteres

---

<sup>3</sup> Los procesos ETL (**por su traducción en español extraer-transformar-cargar del inglés extraxt-transform-load**) son una parte de la integración de datos, siendo elementos importantes cuya función completa el resultado de todo el desarrollo de la cohesión de aplicaciones y sistemas. Este proceso consta precisamente de estas tres fases: extracción, transformación y carga. (PowerData, 2013)

alfabéticos de los demás, en particular del caracter espacio, pero cuando se tratan **datos anagráficos**<sup>4</sup>, es muy común encontrarse con duplicados debidos a errores tipográficos y de **data entry**. (En español entrada de datos).

Existe un compendio de herramientas en **PL/SQL**, capaces de *matchear* los registros y eliminar los duplicados. Para esto se cuenta con la función **SOUNDEX** que devuelve una representación fonética de una cadena, mostrando una codificación de su pronunciación en inglés; aunque esta función puede ser muy útil en cualquier otro idioma. (Rodríguez, 2012)

Según (Rodríguez, 2012), el paquete **UTL\_MATCH** contiene unas funciones muy útiles en el proceso de limpieza de datos (**data cleansing**), desarrolladas para **facilitar la detección de duplicados**. Estas funciones son las siguientes:

- ✓ **EDIT\_SIMILARITY**: calcula la distancia de *Levenshtein*<sup>5</sup>. La distancia se calcula como número de inserciones, cambios y cancelaciones de caracteres que permiten pasar de la cadena 1 a la cadena 2. La función devuelve un número que representa la distancia entre las dos cadenas y el 0 indica dos cadenas idénticas.
- ✓ **EDIT\_DISTANCE\_SIMILARITY**: actúa como la función *EDIT\_SIMILARITY* pero devuelve un valor normalizado entre 100 para las cadenas coincidentes y 0 para la discordancia.
- ✓ **JARO\_WINKLER**: esta función utiliza el algoritmo de *Jaro-Winkler* que calcula un índice de similitud entre dos cadenas, para intentar tener en cuenta posibles errores tipográficos. El valor 1 representa dos cadenas coincidentes.
- ✓ **JARO\_WINKLER\_SIMILARITY**: la función utiliza el algoritmo de *Jaro-Winkler* y normaliza los resultados entre 0 y 100.

---

<sup>4</sup> Registro de datos personales. (Reverso, 2005)

<sup>5</sup> Esta medida debe el nombre al científico ruso que desarrolló un algoritmo para medir la distancia entre dos cadenas de texto s1 y s2

### **1.7 Tecnologías para generar ondas de voz sintética**

Para generar las formas de onda de voz sintética, se necesita hacer uso de dos importantes tecnologías. Las cuales son conocidas como síntesis por concatenación y síntesis de formantes. La **síntesis por concatenación**, se basa en la concatenación de segmentos de voz registrada y aunque produce la voz sintetizada lo más natural posible se perciben las diferencias entre las variaciones naturales en el habla y la naturaleza de las técnicas automatizadas, que para la segmentación de las formas de onda, a veces muestran perceptibles problemas técnicos en la salida. La síntesis por concatenación puede ser implementada de 3 formas fundamentales, las cuales son conocidas como síntesis por selección de unidad, síntesis de *diphone* y síntesis de dominio específico. Por su parte la **síntesis de formantes** crea la salida de voz sintetizada utilizando síntesis aditiva y un modelo acústico. Este método es conocido también como **reglas basadas en síntesis**. (WebAcademia, 2013)

Los sistemas de **síntesis de formantes** de habla sintetizada pueden ser fiables e inteligibles, incluso a velocidades muy altas, evitando los problemas técnicos acústicos que comúnmente afectan a los sistemas de concatenación. Son utilizados por personas con discapacidad visual para navegar rápidamente a los ordenadores que utilizan un lector de pantalla, ya que generalmente son más pequeños que los programas de los sistemas de concatenación, pues no tienen una base de datos de muestras de voz. Dado que los sistemas basados en formantes tienen un control completo de todos los aspectos de la voz de salida, una gran variedad de prosodias y entonaciones se puede emitir y transmitir a preguntas y declaraciones, con variedad de emociones y tonos de voz. (WebAcademia, 2013)

En los sintetizadores de voz, para determinar una correcta pronunciación de las palabras de acuerdo con su ortografía, han surgido distintos tipos de enfoques. El **enfoque basado en diccionario**, es una muestra de ellos, consiste en que un diccionario contiene todas las palabras de un idioma y las pronunciaciones correctas son almacenadas por el programa. La determinación de la pronunciación correcta de cada palabra es una cuestión de buscar cada palabra y situar su ortografía con la pronunciación especificada en el diccionario. Otro conocido enfoque es el **enfoque basado en regla**, que consiste en definir normas de pronunciación y aplicarlas a las palabras para determinar sus pronunciaciones, basado en su ortografía. Esto es similar a la fonética sintética o enfoque a la lectura de aprendizaje. (WebAcademia, 2013)

El enfoque basado en reglas es muy rápido y preciso, pero falla completamente si se le da una palabra que no aparezca en su diccionario y a medida que crece el tamaño del diccionario, también lo hace el espacio requerido en memoria. El enfoque basado en reglas funciona con cualquier entrada de palabras, pero su complejidad crece a medida que el sistema toma en cuenta ortografía reguladora y pronunciaciones. Por lo que un sistema de síntesis de voz, normalmente combina ambos enfoques debido a que las ventajas de uno, contribuye a aminorar las desventajas del otro, trayendo como consecuencia, pronunciación de las palabras, en mejores instantes de tiempo y con más precisiones en su ortografía.

### **1.8 Análisis de las soluciones similares**

En investigaciones realizadas en distintas fuentes de información se ha podido observar que son escasas las herramientas compatibles con los navegadores, que garanticen una salida de audio, factible, intangible y naturalizada. Existen alternativas libres y gratuitas, pero pocas son factibles para su integración con el marco de trabajo Xalix. Dejando así, solo un pequeño conjunto de soluciones, las cuáles son descritas a continuación:

**CLICK Speak:** Lector de pantalla disponible para el navegador Mozilla Firefox, cuenta con voces en inglés y además está disponible para los sistemas operativos *Windows*, *Linux* y *Mac*. (GW Micro, 2014)

**FireVox:** Lector de pantalla disponible para el navegador Mozilla Firefox, cuenta con voces en inglés, español, francés, alemán, italiano y además está disponible para los sistemas operativos *Windows*, *Linux* y *Mac*. (FireVox, 2014)

**VozMe:** Lector de pantalla totalmente gratuito. Cuenta con voces en español, italiano, catalán, hindi e inglés. (Vignoni, 2007) Permite añadir un botón (*bookmarklet*) al navegador, para la sintetización de voz. Está disponible para el navegador Mozilla y para Explorer a partir de su versión 8, como un *plugin*<sup>6</sup> y como

---

<sup>6</sup> Módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande. (masadelante.com, 2014)

programa ejecutable. Es una herramienta destinada a la lectura de sitios web desarrollados con *Wordpress*<sup>7</sup>.

### **Conclusiones parciales**

Por tanto hasta la actualidad han existido una serie de herramientas para leer texto. Todas esas herramientas facilitan el trabajo en la web, principalmente el de usuarios con deficiencias visuales, pero muestran una deficiencia en común; han sido desarrolladas para ser utilizadas solamente con el navegador Firefox Mozilla, en muy pocas ocasiones con versiones de *Internet Explorer* de la 8 en adelante. Además funcionan como un *plugin* de audio de los navegadores, o como un programa externo destinado a su uso en un navegador específico; que el usuario debe adquirir de forma independiente al sitio web por el cuál navegue.

### **1.9 Tendencias y tecnologías**

La aparición de los primeros sintetizadores de voz (*TTS*); la electrónica, y la era digital dieron un gran paso de avance, acercándose mucho más al usuario y facilitando su trabajo con el software. Luego surgieron los lectores de pantallas y con ellos se mostró una gigantesca evolución, en cuanto a las posibilidades que se les brindan a personas con deficiencias visuales, de acceder al ordenador.

Es imperiosa la necesidad de utilizar sintetizadores de voz en el trabajo con lectores de pantallas, para facilitar su manejo y tratamiento. El análisis de las soluciones similares, realizado anteriormente, mostró la existencia de una serie de sintetizadores de voz que pueden ser utilizados para garantizar el funcionamiento de lectores de pantalla, debido a las prestaciones que ofrecen.

### **1.10 Herramientas de síntesis de voz**

A continuación se realizará un análisis de sintetizadores de voz que podrán ser integrados al componente que se desea implementar, para la lectura de pantalla.

---

7 WordPress es una avanzada plataforma semántica de publicación personal orientada a la estética, los estándares web y la usabilidad. WordPress es libre y, al mismo tiempo, gratuito. (Wordpress, 2013)

**FreeTTS:** es un sistema de síntesis de voz realizado en Java, basado en *Flite*, cuya última versión incluye soporte para un número de voces que se encuentren entre [8khz *diphone*, 16khz *diphone*] masculinas y en inglés de Estados Unidos. Importa diversos formatos de voces y tiene soporte parcial para *JSAPI*. Puede ser utilizado para un servidor *TTS*, motor *TTS* para escritorio y como aplicación web descargable FreeTTS. (SentidoWeb, 2007)

**Festival Speech Synthesis System:** es un sistema muy completo hecho para convertir texto en sonido y viceversa. Es de propósito general para múltiples lenguajes. (EIAlecs, 2006)

Se distribuye bajo licencia X11-type permitiendo uso comercial y no comercial sin restricción. Está escrito en lenguaje C++ y está implementado como un intérprete de comandos el cual puede conectarse con diversos módulos y aplicaciones. Además existen librerías para el desarrollo de aplicaciones en los lenguajes Java y C++ así como una interfaz para el editor de textos Emacs. (Universidad de Endinburgh, 2014)

**AT&T:** La primera generación de este sintetizador de voz dependía de la doble concatenación usando codificación de predicción lineal (*LPC*<sup>8</sup> por sus siglas en inglés). *AT&T Natural Voices* está habilitado en tres configuraciones: Servidor, Servidor-Ligero y Escritorio. (Luna, 2007) Las últimas versiones de este sintetizador exigen el uso de licencias comerciales. (AT&T Labs, 2011)

**eSpeak:** Es una herramienta libre, para síntesis de voz. Ofrece una gran variedad de configuraciones para ser aplicadas y obtener mejoras en las voces y en la síntesis. Por otro lado también ofrece una portabilidad entre sistemas operativos como GNU/Linux y Microsoft Windows. Su voz es robótica y es causada por el tipo de síntesis aplicada (basado en formantes). Las voces se crean en 2 archivos *\_list* y *\_rules* (en el caso del español sería *es\_list* y *es\_rules*), el primero contiene una lista de excepciones y atributos para ciertas palabras, y el segundo contiene la pronunciación de los fonemas del idioma haciendo de esta forma el trabajo más engorroso. (MundoAccesible, 2009)

---

<sup>8</sup> Sistemas que tienen un alto grado de inteligibilidad, pero las voces producidas tienden a sonar monótonas y poco naturales. A su vez, permiten un alto grado de control a lo largo del tiempo sobre la frecuencia fundamental, velocidad, volumen y otras características.

### **1.11 Sintetizadores de voz seleccionados**

Por las características que tendrá el componente se ha decidido utilizar dos sintetizadores de voz, el ***Festival Speech Synthesis System*** por ser un *TTS* multilingüe, que funciona correctamente para distribuciones Linux, que es gratis y además haciendo uso de una serie de herramientas, puede ser utilizado en Windows. Cuenta con una serie de voces en castellano que pueden ser cambiadas y utilizadas según lo que se desee realizar, normalización de formantes de difono si se desea probar con licencia comercializadora o para fines militares y de *MBROLA* en cualquier otro caso.

Se integrará además el sintetizador de texto a voz ***eSpeak***, permitiendo comprobar la usabilidad del componente y su capacidad de integrarse fácilmente con otros sintetizadores. Además este sintetizador tiene características muy parecidas a las de ***Festival Speech Synthesis System***, difiriendo particularmente en la calidad de las voces.

### **1.12 Metodologías de desarrollo de software**

Un proceso de desarrollo es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. (Jacobson, et al., 2000) Ver la figura 1. Debido a la abstracción y complejidad que tienen los productos que solicitan los usuarios, se dificulta el trabajo de analistas y desarrolladores, y a su vez el proceso de desarrollo se vuelve muy complejo. Surge la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto (software). Esto es conocido como metodología de desarrollo. Se clasifican en dos tipos: metodologías ágiles y metodologías tradicionales. Las más utilizadas a nivel mundial son: La Programación Extrema (*XP*), Scrum y el Proceso Unificado de Software (*RUP*), las cuales serán estudiadas seguidamente.



**Figura 1. Proceso de desarrollo de software**

### 1.12.1 Metodologías tradicionales

Según (Jacobson, et al., 2000) las metodologías tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, en las herramientas a utilizar y requiere una extensa documentación, ya que pretenden prever todo de antemano. Este tipo de metodologías son más eficaces en proyectos a largo plazo y que requiera gran cantidad de recursos. Una de las metodologías tradicionales más conocidas y utilizadas es la Metodología RUP (Proceso Unificado de *Rational*).

#### Proceso Unificado de *Rational* (RUP en lo adelante)

##### Características de RUP

RUP es una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo. Es un método pesado, en el cual un cambio en las etapas de vida del sistema incrementaría notablemente el costo. Define las actividades y proporciona los artefactos necesarios para construir el software. Es el proceso de desarrollo más general de los existentes actualmente. Realiza un levantamiento exhaustivo de requerimientos. Busca detectar defectos en las fases iniciales. Intenta reducir al número de cambio y realizar el análisis y diseño tan completos como sea posible y se basa en el diseño genérico e intenta anticiparse a futuras necesidades. (Flores, 2014)

### 1.12.2 Metodologías ágiles

Según (Solís, 2003) las metodologías ágiles se centran en valorar al individuo y las iteraciones del equipo más que a las herramientas o los procesos utilizados. Es más importante crear un producto de software

que funcione que escribir mucha documentación. Sus fundamentos se basan en que el cliente está en todo momento colaborando en el proyecto y que es más importante la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan.

### **Programación Extrema (XP en lo adelante)**

#### **Características**

*XP* busca simplificar el desarrollo del software y lograr reducir el costo del proyecto. El desarrollo de software es riesgoso y difícil de controlar. Se rediseñará todo el tiempo, dejando el código siempre en el estado más simple posible. Las iteraciones serán más cortas de lo que es usual en otros métodos, esto permite beneficiarse de la retroalimentación tan a menudo como sea posible. Intenta reducir la complejidad del software por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. Falla cuando se utiliza en dominios de aplicaciones complejas o situaciones difíciles en la organización, además generalmente no es utilizada en proyectos de larga duración. (Flores, 2014)

#### **Scrum**

#### **Características**

Scrum busca el cumplimiento de expectativas del cliente, estimando el valor que le aporta cada requisito del proyecto y estableciendo su prioridad. Muestra flexibilidad a cambios, ya que está diseñado para adaptarse a los cambios de requerimientos que conllevan los proyectos complejos. La reducción del tiempo de *marketing* permite que el cliente utilice las funcionalidades más importantes del proyecto antes de que esté finalizado por completo. Define 3 fases conocidas como pre-juego, juego y post-juego y se divide en 4 flujos de trabajo. (Softeng, 2014)

### **1.13 Metodología seleccionada**

De acuerdo a las características que presenta el componente que se va a desarrollar, se ha decidido utilizar *RUP*. *XP* y Scrum son para proyectos de corto plazo y se basan en iteraciones pequeñas. Como tratan de

reducir el costo del proyecto y de simplificar el desarrollo de software, generan muy poca documentación, muy útil para el desarrollo de proyectos a largo plazo y necesaria tanto para todo el personal del proyecto como para el cliente, lo cual no es factible en el caso del desarrollo del componente, puesto que es su versión 1.0, y se requiere un cantidad de documentación y procesos de la ingeniería que favorezcan la implementación. *RUP* es para proyectos y equipos grandes y no requiere la presencia permanente del cliente, durante todo el proceso de desarrollo. Esta última es una metodología robusta y bien definida que entre sus principales características se encuentra que permite llevar a cabo un proceso de desarrollo práctico, brindando amplias guías y plantillas para el desarrollo de software. Lo anterior expuesto conllevó a la toma de decisión.

#### **1.14 Herramientas CASE<sup>9</sup> (Ingeniería de software asistida por computadoras, en español)**

Las herramientas *CASE* son un complemento de la caja de herramientas del ingeniero de software. *CASE* proporciona al ingeniero la posibilidad de automatizar actividades manuales y de mejorar su visión general de la ingeniería. Al igual que las herramientas de ingeniería y de diseño asistidos por una arquitectura de entorno, abarca un hardware y un software de sistema adecuados. Además, la arquitectura del entorno debe considerar los patrones de trabajo humano que se aplicarán durante el proceso de ingeniería de software. (Pressman, 1998)

Según (Gómez, 2014), las herramientas *CASE* se clasifican en diferentes clases, de acuerdo a una serie de características que deben cumplir.

✓ **Clasificación de acuerdo a su amplitud:**

1. **TOOLKIT** colección de herramientas integradas que permiten automatizar la planificación estratégica, el análisis, diseño y generación de programas presentes en las fases del ciclo de vida del sistema informático.

---

<sup>9</sup> Del inglés Computer Aided Software Engineering

2. **WORKBENCH** conjuntos integrados de herramientas que dan soporte a la automatización del proceso completo de desarrollo del sistema informático. Permiten cubrir el ciclo de vida completo. El producto final aportado por ellas es un sistema en código ejecutable y su documentación.
- ✓ **Clasificación de acuerdo a las fases del ciclo de vida:**
1. **UPPER CASE** planificación estratégica, requerimientos de desarrollo funcional de planes corporativos.
  2. **MIDDLE CASE** análisis y diseño.
  3. **LOWER CASE** generación de código, test e implantación.

### **Lenguaje de modelado unificado (en lo adelante UML)**

*UML* es un popular lenguaje de modelado de sistemas de software. Es un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software. (Diccionario de informática, 2014) Captura decisiones y conocimiento sobre los sistemas que se deben construir y se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre ellos. Está diseñado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. (Jacobson, et al., 2000)

#### **1.15 Herramientas CASE para el modelado UML**

**Microsoft Project** es una herramienta que está disponible para los sistemas operativos Windows, DOS, Mac, es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas,

dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo. Es útil para la gestión de proyectos, aplicando procedimientos descritos en el *PMBok*<sup>10</sup> del *PMI*<sup>11</sup>. Permite el seguimiento de las tareas y recursos, comparar versiones de planes de proyectos, evaluar los cambios, realizar un seguimiento del rendimiento, generar informes predefinidos, compartir planes de proyecto, colaboración entre grupos de trabajo, presenta diagramas como el de Grant y el de Pert (diagrama de red). (Alonso, 2009)

**Rational Rose** es una herramienta de producción y comercialización establecidas por *Rational Software Corporation* (actualmente parte de *IBM*). Utiliza el lenguaje *UML* como medio para facilitar la captura de dominio de la semántica, la arquitectura y el diseño. No es gratuita, por lo que se debe hacer un previo pago para poder adquirir el producto. Habilita asistentes para crear clases y provee plantillas de código que pueden aumentar significativamente la cantidad de código fuente generada. Está disponible para los sistemas operativos Windows y Linux. (Alonso, 2009)

**JDeveloper** es un magnífico entorno integrado desarrollado por Oracle que trabaja con la ingeniería inversa<sup>12</sup>. Es un software propietario pero gratuito desde 2005. Es desarrollado para Java y posee diagrama de clases (*UML*). Está disponible para los sistemas operativos Windows, Linux y Mac OSX. (Alonso, 2009)

**MagicDraw** es una de las mejores herramientas **CASE** para el modelaje *UML*. Facilita el análisis y diseño de los sistemas y de las bases de datos orientados objeto. Facilita el diseño con auto completamiento y corrección automática en tiempo real, posibilita la derivación de modelos *UML* a través de códigos fuente escritos anteriormente. Además facilita y agiliza el cambio del dominio del modelado, genera

---

<sup>10</sup> Es un estándar reconocido internacionalmente (IEEE, ANSI) este trabaja con el uso del conocimiento, de las habilidades, de las herramientas, y de las técnicas para resolver requisitos del proyecto. (MurdenManX, 2010)

<sup>11</sup> Es una asociación profesional sin fines de lucro más importante y de mayor crecimiento a nivel mundial que tiene como misión convertir a la gerencia de proyectos como la actividad indispensable para obtener resultados en cualquier actividad de negocios. (ADDKW, 2012)

<sup>12</sup> La ingeniería inversa es el proceso que consistente en analizar un programa, realizando un esfuerzo por crear una representación del programa con un nivel de abstracción más elevado que el del código fuente. La ingeniería inversa es un proceso de recuperación de diseño. (Pressman, 1998)

automáticamente los informes, está disponible para los sistemas operativos Linux y Windows y es una herramienta gratuita. (Alonso, 2009)

**Visual Paradigm** es una herramienta que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas, contando con un enfoque orientado a objeto. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades, pero también tiene licencia gratuita y comercial. Además es multiplataforma. Se caracteriza por tener un diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad, el uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación, sus capacidades de ingeniería directa e inversa y la sincronización del modelo y código en todo el ciclo de desarrollo. Esta herramienta soporta aplicaciones Web, genera código para Java y exporta como HTML diagramas de procesos de negocio, decisión y actor de negocio. Cuenta con un modelado colaborativo con CVS y Subversión (control de versiones). Permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y mantenimiento del software. Facilita la reutilización del software, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería de Software. (Pressman, 1998)

### **1.16 Selección de una herramienta CASE para el modelado UML**

Luego de realizar un detallado estudio sobre herramientas CASE para el modelado UML, se decidió que la más adecuada para el equipo de trabajo debe ser el *Visual Paradigm*. Entre las características que se analizaron e influyeron en esta decisión se encuentran las facilidades que brinda para el trabajo y modelado de casos de uso y que soporta la versión 2.0 de UML. También se tuvo en cuenta que es una herramienta que ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas, contando con un enfoque orientado a objeto. Se analizó su posibilidad de soportar aplicaciones Web y que exporta como HTML distintos diagramas. Facilita la reutilización del software, portabilidad y estandarización de la documentación y usa distintas metodologías propias de la Ingeniería de Software.

### 1.17 Selección del lenguaje a utilizar

Debido a que se trabajará sobre el marco de trabajo Xalix, se deben utilizar los lenguajes de programación, tanto del lado del cliente o como del servidor que define el mismo, estos son:

1. **JavaScript** para el manejo de datos ya que es un lenguaje de programación interpretado, que se utiliza principalmente para crear páginas web dinámicas.
2. **CSS 3** para el estilo debido a que es un lenguaje que mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes y se utiliza para definir el aspecto de todos los contenidos.
3. **HTML5** ya que mejora el posicionamiento. Permite también el trabajo con video y audio sin necesidad de contar con herramientas externas, solo es necesario que el navegador soporte HTML5.
4. **PHP 5.3** ya que es un lenguaje para el desarrollo de aplicaciones web dinámicas que se puede incorporar directamente a páginas HTML. Es **portable** y permite utilizarlo en cualquier servidor web sin importar el sistema operativo. Es un lenguaje robusto y estable que dota de gran seguridad a las páginas web contra ataques no deseados. Tiene una gran capacidad para conectarse a cualquier base de datos incluida PostgreSQL.

### 1.18 Framework para el desarrollo

El *framework* es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para desarrollar una aplicación concreta. (Carrasco, et al., 2009)

#### Capa lógica del negocio

Debido a que se trabajará sobre el marco de trabajo Xalix, se debe utilizar el framework de desarrollo que el mismo define; este es Symfony 2.3.4

**Symfony 2.3.4:** Symfony 2 es la versión más reciente de Symfony, el framework para desarrollar aplicaciones PHP supone un cambio radical tanto en arquitectura interna como en filosofía de trabajo respecto a sus versiones anteriores. Su arquitectura interna está completamente desacoplada, lo que permite reemplazar o eliminar fácilmente aquellas partes que no encajan en el proyecto. (Pérez, 2013)

### **Capa de acceso a datos**

**Doctrine 2.3:** Doctrine2 es un mapeador relacional de objeto (ORM) para PHP 5.3.0+ que proporciona persistencia transparente de objetos PHP. Se sitúa en la parte superior de una poderosa capa de abstracción de base de datos (DBAL por *DataBase Abstraction Layer*). La principal tarea de los ORM es la traducción transparente entre objetos (PHP) y las filas relacionales de la base de datos. Una de las características clave de Doctrine es la opción de escribir las consultas de base de datos en un dialecto SQL propio orientado a objetos llamado Lenguaje de Consulta Doctrine (DQL por *Doctrine Query Language* en inglés), inspirado en *Hibernates HQL*. (Pacheco, 2011) Según (Eberlei, 2012), Doctrine2 incluye nuevas versiones de los paquetes Común, DBAL y ORM. Además facilita la compatibilidad hacia atrás con cada versión anterior tanto como sea posible.

### **1.19 Sistema gestor de base de datos**

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar las siguientes acciones:

- ✓ Definición de los datos.
- ✓ Mantenimiento de la integridad de los datos dentro de la base de datos.
- ✓ Control de la seguridad y privacidad de los datos.
- ✓ Manipulación de los datos.

### **1.20 Selección de un gestor de base de datos:**

Debido a que se trabajará sobre el marco de trabajo Xalix, se debe utilizar el gestor de base de datos que el mismo define; este es PostgreSQL v 9.3

**PostgreSQL 9.3:** Según (CJava, 2013) intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle y bajo la licencia BSD, la cual concede ciertas libertades adicionales, entre ellas hacer software propietario. PostgreSQL ofrece una garantía de integridad en los datos mucho más fuerte que MySQL. PostgreSQL presenta una mejor escalabilidad y rendimiento cuanto más grandes sean las cargas de trabajo. Tiene muy buen soporte para *triggers* y procedimientos en el servidor. Además soporta un subconjunto de SQL92 mayor que el que soporta MySQL. La versión 9.3 de PostgreSQL amplía su fiabilidad, disponibilidad y capacidad de integración con otras bases de datos, permitiendo crear aplicaciones que no habrían sido posible antes.

### **1.21 Entorno de desarrollo integrado (IDE en lo adelante)**

Según (García, 2013) un *IDE*, es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi y Visual Basic.

**NetBeans IDE 7.3.1:** es de código abierto disponible para Windows, Mac, Linux y Solaris, que permite a los desarrolladores crear aplicaciones web, empresariales, de escritorio y móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript, Ajax, Ruby y Ruby on Rails, Groovy and Grails y C/C++. (Netbeans, 2014)

**Zend Studio 10.6:** sirve de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. Cuenta con versiones para Windows, Linux y MacOS. Zend Studio consta de dos partes que se instalan por separado. Las del cliente contando con la interfaz de edición y ayuda; y las del servidor que instalan Apache y el módulo PHP. (Álvarez, 2003)

**PHP Designer 8.0.0:** es un editor de texto para programadores de PHP. Satisface casi todas las necesidades de codificación de PHP y su principal objetivo es aumentar la productividad, facilitando la tarea. La ventaja principal consiste en el resaltado de sintaxis inteligente, que marca errores a medida que escribe.

## 1.22 Selección de un IDE

Se eligió el NetBeans IDE 7.3.1 ya que soporta los lenguajes de la web, tanto los del lado del cliente (HTML, CSS, JavaScript) como los del servidor (en este caso PHP), y se integra fácilmente con los *frameworks* escogidos para el desarrollo (Symfony, jQuery y Doctrine). También cuenta con un sistema de proyectos basado en control de versiones y refactorización, además de ser gratuito y sin prohibiciones de uso.

## 1.23 Servidor web

Según (Gregorio, 2011) un servidor web o servidor http es un programa que procesa cualquier aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. Se mantiene a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. Existen distintas clasificaciones de servidores web:

- ✓ **Servidor dedicado:** se refiere a una computadora servidora dedicada exclusivamente al sitio del cliente (para aplicaciones de alta demanda).
- ✓ **Servidor compartido:** un mismo servidor (computadora + programa servidos) se usará para varios clientes compartiendo los recursos.

Según (Gregorio, 2011) los **servidores web más utilizados son:**

**Apache 2.4.7:** lo caracterizan su estabilidad, confiabilidad y el gran aporte del grupo de voluntarios que planean y desarrollan todo lo relativo a esta plataforma, desde la documentación hasta el mismo código en sí. Además, es gratuito, de código abierto y corre sobre cualquier plataforma. (Gregorio, 2011)

**Lighttpd:** está pensado para hacer cargas pesadas sin perder balance, utilizando poca RAM y poca de CPU. Entre sus principales características se encuentran seguridad, rapidez, cumplimiento y flexibilidad. Está diseñado y optimizado para entornos de alto rendimiento. Con una pequeña huella de memoria en comparación con otros servidores web, la gestión eficaz de la CPU en vacío, y el conjunto de características avanzadas. Es de código abierto bajo licencia BSD revisada para web 2.0. (Lighttpd, 2014)

#### **1.24 Selección del servidor Web a utilizar**

El estudio anterior permitió decidir que se utilizará como servidor web Apache 2.0. Las características y ventajas que permitieron tomar la decisión son: cuenta con una gran estabilidad, confiabilidad. Tiene licencia de software libre y de código abierto. Su seguridad, lo hace especial respecto a otros servidores como Lighttpd que tienen una mayor probabilidad de contener vulnerabilidades.

#### **1.25 Conclusiones**

El estudio realizado permitió identificar las funcionalidades que se tendrán en cuenta para la posterior implementación del complemento para integrar lectores de pantalla con el marco de trabajo Xalix. Para el desarrollo de la propuesta de solución se decidió que se utilizarían herramientas, en concordancia con las características expuestas y explicadas con anterioridad. Estas herramientas quedaron definidas de la siguiente manera, JavaScript para el manejo de datos, CSS para el estilo, HTML5 para la estructura y contenido, PHP 5.3 como lenguaje de programación. Los frameworks seleccionados Symfony 2.3.4 y Doctrine 2.3, Netbeans 7.3.1 como IDE, Apache 2.4.7 como servidor web, la metodología RUP y el lenguaje de modelado UML 2.0, la herramientas CASE Visual Paradigm 8.0 para la modelación de los diagramas y como gestor de base de datos PostgreSQL 9.3.

## Capítulo 2: Características del componente

### 2.1 Introducción

El **modelo de dominio** forma parte del flujo de trabajo de **modelado del negocio**. En el presente capítulo se realiza un **modelo de dominio**, en función de lograr un mejor entendimiento del sistema durante el proceso de desarrollo del software. Luego se realiza un levantamiento de los requisitos funcionales y no funcionales del sistema, para determinar con claridad las capacidades y cualidades que el producto debe poseer. Posteriormente se identifican los **actores** y los **casos de uso** para realizar el diagrama de casos de uso del sistema.

### 2.2 Modelo del dominio

Según (Jacobson, et al., 2000), un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Debido a lo antes mencionado se decide realizar un modelo de dominio con el fin de determinar los conceptos más importantes asociados.

#### 2.2.1 Conceptos del dominio

Las plataformas educativas contarán con un componente, que contribuirá a la accesibilidad de los contenidos para los usuarios con deficiencias visuales. El componente consiste en un lector de pantalla encargado de interpretar textos y reproducirlos. El lector de pantalla, requiere un sintetizador de voz para normalizar el texto, elaborar una prosodia intangible y convertirlo en un archivo de audio, el cual será reproducido de forma transparente al usuario a través del navegador web que utilice.

A continuación se describen los conceptos principales del dominio, asociados a la descripción del problema existente:

**Usuario:** Persona que accede a la plataforma educativa.

**Plataforma educativa:** Es un software educativo destinado a apoyar el proceso de enseñanza-aprendizaje.

**Lector de pantalla:** Es una herramienta asistida que identifica e interpreta lo que se muestra en la pantalla de un ordenador mediante voz.

**Sintetizador de voz:** Es la tecnología que permite convertir el texto a audio.

**Normalizador de texto:** Herramienta encargada de asignar un componente léxico coherente a cada palabra para crear un texto entendible por el sintetizador de voz.

### 2.2.2 Modelo del dominio

El modelo del dominio que se muestra a continuación, representa la relación existente entre los conceptos identificados anteriormente. Los mismos se definen mediante clases.

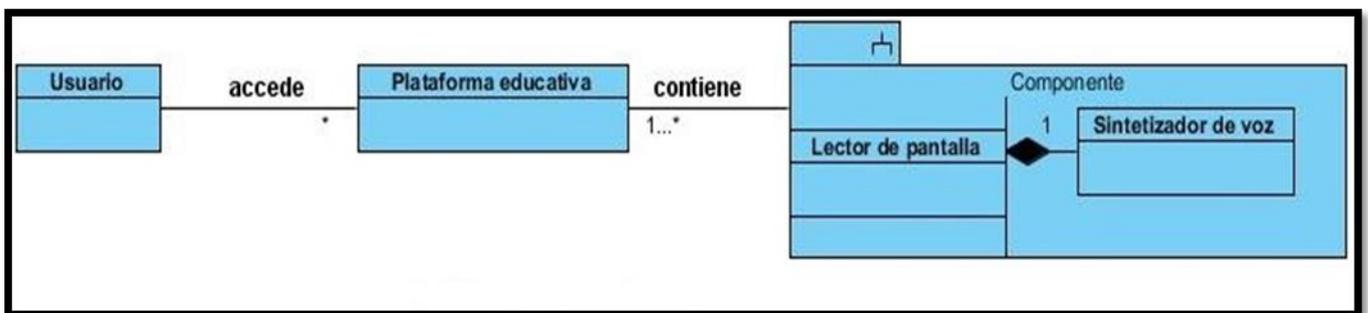


Figura 2. Modelo del dominio

### 2.3 Descripción de las funcionalidades propuestas

Las funcionalidades propuestas se encontrarán integradas con el componente, al cual la plataforma educativa le enviará la petición realizada por el usuario, con el fin de satisfacerla. El componente está formado por un lector de pantalla y un sintetizador de voz. A continuación se describe su funcionamiento.

**Componente:** El componente es el encargado de facilitar la comunicación entre el lector de pantalla y la web. Además, en esta sección se adicionan nuevos sintetizadores de texto a voz. Su correcto funcionamiento, depende de la instalación e implementación de las clases encargadas de este proceso.

Para saber los pasos que se deben seguir en la instalación del componente, remitirse al **Anexo # 1 Instalación del componente**. Para saber los pasos que se deben seguir para adicionar nuevos sintetizadores de texto al sistema y que estas funciones como es debido, remitirse al **Anexo # 2 Implementación e integración de nuevos sintetizadores de texto a voz al componente**.

**Lector de pantalla:** En esta sección se obtiene el texto, se le envía al sintetizador de voz y luego reproduce el archivo de audio devuelto por el sintetizador de voz.

**Sintetizador de voz:** En esta sección se captura el texto enviado por el lector de pantalla, se normaliza y luego se sintetiza, devolviendo un archivo de audio al lector de pantalla.

## **2.4 Requisitos**

Según (Jacobson, et al., 2000) se denomina captura de requisitos al proceso de averiguar, normalmente en circunstancias difíciles, lo que se debe construir. A continuación se realizará la especificación de los requisitos tanto funcionales como no funcionales, con el fin de definir las características y capacidades que debe tener el componente.

### **2.4.1 Requisitos funcionales**

Los requisitos funcionales especificados muestran las capacidades y funcionalidades que debe tener el sistema. Deben ser claros, precisos y posibles de verificar.

**RF1: Capturar texto seleccionado por el usuario.**

**RF2: Normalizar texto.**

**RF2.1:** El sistema debe permitir que el texto sea normalizado a través de un sintetizador específico. (Ejemplo utilizando el sintetizador *Festival Speech Synthesis System*).

**RF2.2:** El sistema debe permitir el uso de una herramienta externa para normalizar texto (para el caso en el que no se utilice un sintetizador de voz que no normalice texto internamente).

**RF3: Permitir la síntesis a voz del texto a través de herramientas encargadas de este proceso.**

**RF4: Cambiar voz del sintetizador de voz, siempre que este las tenga disponibles.**

**RF5: Configurar opciones provistas por el sintetizador de voz disponibles; solo para el administrador del sistema.**

**RF5.1** El sistema debe permitir adicionar sintetizadores de texto.

**RF5.2** El sistema debe permitir editar los sintetizadores de texto existentes.

**RF5.3** El sistema debe permitir eliminar sintetizadores de texto.

**RF5.4** El sistema debe permitir adicionar voz a los sintetizadores de texto.

**RF5.5** El sistema debe permitir editar voz de los sintetizadores de texto.

**RF5.6** El sistema debe permitir eliminar voz de los sintetizadores de texto.

**RF6: Reproducir texto.**

**RF6.1:** El sistema debe contar con un botón para sintetizar texto, que se encargue de iniciar la reproducción de texto.

**RF6.2:** El sistema debe reproducir texto teniendo en cuenta la voz seleccionada.

**RF6.3:** El sistema debe reproducir el texto seleccionado por el usuario.

**RF6.4:** El sistema debe reproducir todo el texto si el usuario inicia la reproducción sin seleccionar un texto previo.

**RF6.5:** El sistema debe permitir el uso de atajos del teclado para reproducir texto.

**RF6.6:** El sistema debe permitir que el usuario detenga la reproducción del texto.

**RF6.7:** El sistema debe permitir que el usuario cambie de un área a otra durante la reproducción de todo el contenido del sitio web, haciendo uso de atajos del teclado.

## **RF7: Descargar archivo de audio.**

### **2.4.2 Requisitos no funcionales**

Los requisitos no funcionales especificados muestran las propiedades y cualidades que debe tener el sistema. Los cuales posibilitan la integralidad, usabilidad, rapidez y confiabilidad. También es necesario definir requerimientos de programas que es necesario instalar para que el sistema de software funcione adecuadamente.

#### **Requerimientos de soporte**

- ✓ Ser generado en tecnología Web para ser accesible a través de Internet.
- ✓ Compatibilidad con los navegadores: Mozilla Firefox, Opera y Chrome.

#### **Restricciones de diseño e implementación**

- ✓ Las tecnologías y lenguajes utilizados para la integración del complemento implementado en la máquina del servidor son los siguientes: Symfony 2.3.4, PostgreSQL 9.3, PHP 5.3.4, JQuery 1.10, HTML5, servidor web apache 2.4.7, como sistemas operativos Linux y Windows.

#### **Requerimientos de usabilidad**

- ✓ Los sitios web que vayan a utilizar el componente, deben cumplir con la normativa definida por la (W3C, 2014) (W3C, 2010) que especifica que para un buen funcionamiento de los lectores de pantalla, es necesario utilizar los roles área, ejemplo para el menú *role= navigation*, para el *body*, *role=main*. Para más información sobre las etiquetas área que se necesitan remitirse al **Anexo # 3 Etiquetas áreas para el funcionamiento del lector de pantalla.**

#### **Requerimientos de portabilidad**

El sistema debe ser multiplataforma, pudiéndose ejecutar con cambios mínimos en el código sobre Microsoft Windows y GNU/Linux.

#### **Requerimientos legales**

El sistema debe ser implementado utilizando herramientas libres.

## 2.5 Actores y casos de uso del sistema

En esta sección se identifican y describen los actores del sistema, se realiza el diagrama de casos de uso del sistema así como la descripción de los casos de usos definidos.

### 2.5.1 Descripción de los actores del sistema

Los actores que se identificaron son personas encargadas de inicializar los casos de uso del sistema e interactuar directamente con ellos.

**Tabla 1. Actores del sistema**

Actor	Descripción
Usuario	Es el usuario que accede a un sitio web realizado bajo el marco de trabajo Xalix, para realizar determinadas operaciones.
Administrador	Es el actor que puede realizar tareas de administración, relacionadas con la configuración del sistema y además realizar las mismas acciones que el usuario.

### 2.5.2 Diagrama de casos de uso del sistema

A continuación se muestra el **diagrama de casos de uso** del sistema que contiene los casos de uso generados a partir de los requisitos identificados anteriormente, así como las relaciones que se establecen entre ellos. Se identificaron 7 casos de uso, basados en los requisitos funcionales, los cuales se muestran a continuación:

- ✓ Cambiar voz del sintetizador de voz
- ✓ Reproducir texto

- ✓ Capturar texto
- ✓ Normalizar texto
- ✓ Sintetizar a voz el texto
- ✓ Descargar archivo de audio
- ✓ Configurar sintetizador de voz

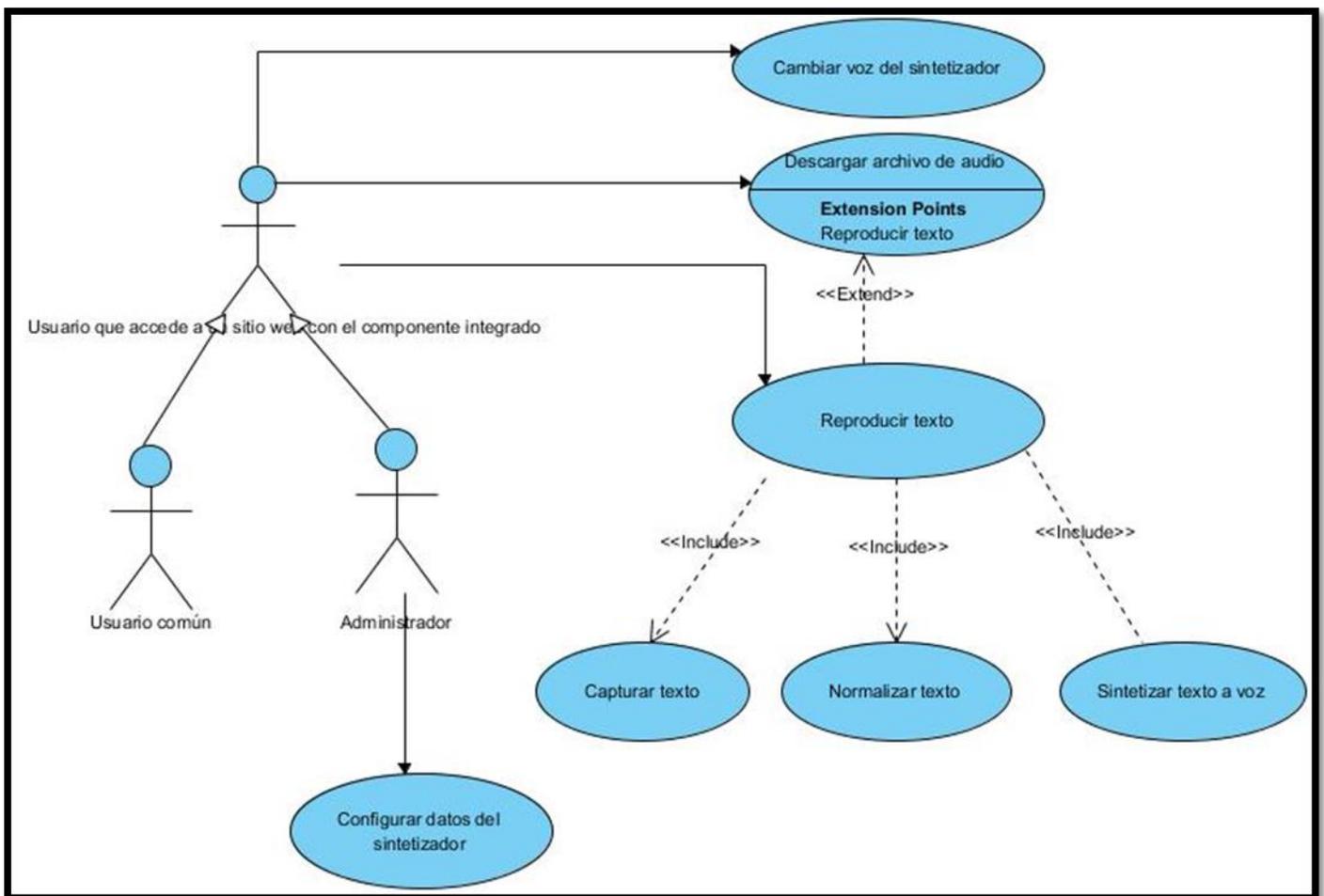


Figura 3. Diagrama de casos de uso

### 2.5.3 Descripción de los casos de uso del sistema

Los casos de uso identificados detallan la secuencia de interacciones que se llevan a cabo entre un actor y el sistema. Permiten establecer acuerdos entre clientes y desarrolladores sobre lo que deben cumplir las funcionalidades.

A continuación se presenta la descripción del caso de uso **Reproducir texto**. Para el estudio de las restantes descripciones remitirse al **Anexo # 4: Descripciones de los casos de uso**.

**Tabla 2. Descripción del CU Reproducir texto**

CU Reproducir texto	
<b>Objetivo</b>	Reproducir el contenido de un sitio web, ya sea solo un texto seleccionado, o el contenido completo. Permitir el uso de atajos del teclado para la reproducción de texto. Además se podrá detener o adelantar la reproducción del texto.
<b>Actores</b>	<ul style="list-style-type: none"> <li>✓ Usuario del sistema (Usuario)</li> <li>✓ Administrador del sistema (Administrador)</li> </ul>
<b>Resumen</b>	El caso de uso inicia cuando el actor del sistema define el texto a reproducir. Una vez reproducido el texto, finaliza el caso de uso.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Alta
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>✓ El usuario accede a un sitio web que tenga instalado el componente</li> <li>✓ El usuario define el texto a reproducir</li> </ul>

<b>Postcondiciones</b>		Se reproduce el texto.
<b>Flujos de eventos</b>		
<b>Flujo básico “Seleccionar voz”</b>		
	<b>Actor</b>	<b>Sistema</b>
1	El actor del sistema despliega la lista de voces.	
2		El sistema muestra una interfaz que permite seleccionar una voz provista por el sintetizador de voz utilizado.
3	El actor del sistema selecciona una de las voces disponibles.	
4		<ul style="list-style-type: none"> <li>✓ El sistema identifica la voz seleccionada</li> <li>✓ El sistema captura y toma esa voz para la posterior reproducción de texto</li> </ul>
<b>Flujos alternos</b>		
1. a El usuario no selecciona voz.		
	<b>Actor</b>	<b>Sistema</b>
1		Tiene una voz en idioma español asignada por defecto.
<b>Sección 1 “Reproducir texto seleccionado”</b>		
<b>Flujo básico “Reproducir texto seleccionado”</b>		
	<b>Actor</b>	<b>Sistema</b>

1	El actor del sistema selecciona un texto.	
2		<ul style="list-style-type: none"> <li>✓ El sistema captura el texto seleccionado por el usuario</li> <li>✓ El sistema muestra una interfaz con un botón para iniciar reproducción</li> </ul>
3	El actor del sistema presiona el botón de sintetizar.	
4		El sistema inicia la reproducción del texto.
<b>Flujos alternos</b>		
5.a El actor no selecciona el texto a reproducir		
	<b>Actor</b>	<b>Sistema</b>
1		El sistema reproduce todo el contenido del sitio web.
<b>Sección 2 “Uso de atajos del teclado para reproducir texto”</b>		
<b>Flujo básico “Uso de atajos del teclado para reproducir texto”</b>		
	<b>Actor</b>	<b>Sistema</b>
1	El actor del sistema selecciona texto y presiona <b>Ctrl + ,</b>	
2		El sistema captura el texto e inicia su reproducción
3	El actor del sistema presiona <b>Ctrl+,</b>	
4		El sistema construye un árbol de navegación a partir de roles

**CAPÍTULO 2 | UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS | FACULTAD 4**  
**COMPONENTE PARA LA INTEGRACIÓN DE UN LECTOR DE PANTALLA AL MARCO DE TRABAJO XALIX**

		áreas definidos en el sitio e inicia la reproducción de todo el contenido relevante
5	El actor selecciona un texto y presiona <b>Ctrl +</b> .	
6		El sistema captura el texto seleccionado y envía la petición de descarga al navegador, que se encargará del resto en dependencia de su configuración.
7	El actor del sistema presiona <b>Ctrl+.</b>	
8		El sistema captura los roles áreas definidos en el sitio y envía la petición de descarga al navegador, que se encargará del resto en dependencia de su configuración.

**Flujos alternos**

5.a El usuario presiona atajo del teclado no válido para el componente

	<b>Actor</b>	<b>Sistema</b>
1		El sistema no hace nada

**Sección 3 “Detener reproducción del texto”**

**Flujo básico “Detener reproducción de texto”**

	<b>Actor</b>	<b>Sistema</b>
1	El actor del sistema presiona la barra espaciadora.	

2		<ul style="list-style-type: none"> <li>✓ El sistema detecta la tecla presionada</li> <li>✓ El sistema comprueba que sea la barra espaciadora</li> <li>✓ El sistema detiene la reproducción del texto</li> </ul>
<b>Flujos alternos</b>		
1. a El actor del sistema presiona una tecla diferente.		
	<b>Actor</b>	<b>Sistema</b>
1		El sistema no detiene la reproducción.
<b>Sección 4 “Adelantar la reproducción”</b>		
<b>Flujo básico “Adelantar la reproducción”</b>		
	<b>Actor</b>	<b>Sistema</b>
1	El actor del sistema presiona <b>Ctrl+*</b>	
2		El sistema verifica que esté realizando la reproducción de todo el contenido del texto, y se desplaza hacia el siguiente bloque de texto en el árbol de navegación
3	El actor del sistema presiona <b>Ctrl+/**</b>	
4		El sistema verifica que esté realizando la reproducción de todo el contenido del texto, y se desplaza hacia el bloque de texto anterior en el árbol de navegación
5	El actor hace uso de las facilidades que brinda la barra de reproducción	

	para desplazar el tiempo de reproducción	
6		El audio se desplaza hacia adelante o hacia atrás en dependencia de la acción del actor
<b>Flujos alternos</b>		
1.a El actor del sistema presiona una tecla distinta a la de adelantar o retroceder		
	<b>Actor</b>	<b>Sistema</b>
1		El sistema continúa la reproducción de texto.
<b>Relaciones</b>	<b>CU Incluidos</b>	<ul style="list-style-type: none"> <li>✓ <b>CU Uso de atajos del teclado</b></li> <li>✓ <b>CU Descargar archivo de audio</b></li> </ul>

## 2.6 Conclusiones

Esta propuesta de solución sirvió para alcanzar una mejor comprensión del sistema a desarrollar. Debido a la baja estructuración de los procesos del negocio se definieron los principales conceptos y se relacionaron en el modelo de dominio. Se seleccionaron los requisitos funcionales y no funcionales, permitiendo así determinar las capacidades y cualidades que el sistema debe cumplir. Se identificaron los actores y los casos de uso del sistema que quedaron plasmados en el diagrama de casos de uso del sistema. Por último se describieron los casos de uso del sistema para un mejor entendimiento de las funcionalidades; permitiendo así la finalización de este flujo de trabajo haciendo posible transitar hacia la construcción de la solución propuesta.

## Capítulo 3: Análisis y diseño del componente

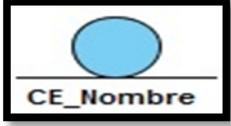
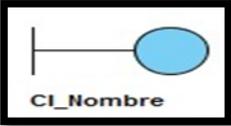
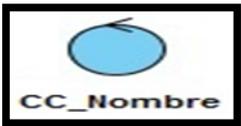
### 3.1 Introducción

Análisis y Diseño es el tercer flujo de trabajo que propone la metodología RUP. Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar. Consiste en transformar los requisitos en un diseño del sistema en creación evolucionando en una arquitectura sólida para el sistema y adaptando el diseño para que se ajuste al entorno de implementación, con un diseño pensado para el rendimiento. (Jacobson, et al., 2000) Por lo que en el presente capítulo se realiza el modelo del análisis, donde se modelan los diagramas de clases de análisis y un diagrama de clases por cada caso de uso. Se realiza además el modelo del diseño donde se efectúa por cada caso de uso un diagrama de clases con estereotipos web. Se realizan los diagramas de iteración, el diagrama entidad relación de la base de datos y se describen todas las tablas. También se definen el patrón arquitectónico, los patrones de diseño a emplear y se realiza el diagrama de despliegue definido correctamente para dar paso a la implementación.

### 3.2 Modelo de análisis

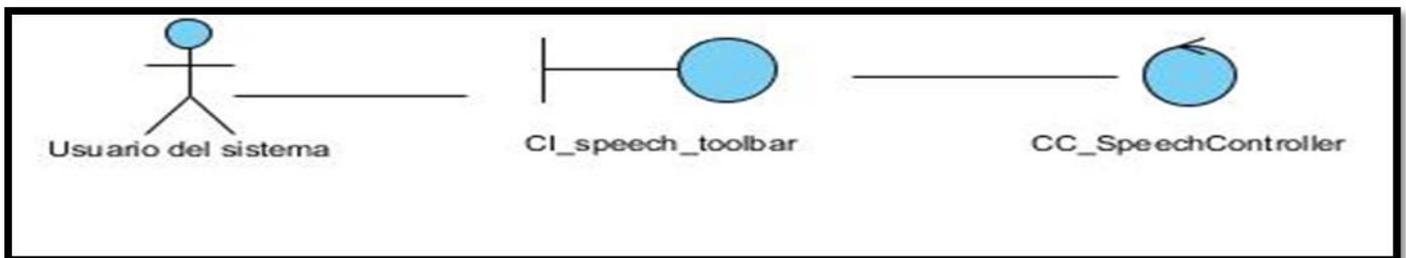
El modelo de análisis proporciona un “diseño preliminar” para un conjunto de requerimientos, se describe en un lenguaje para los desarrolladores y proporciona una visión general y conceptual del sistema respecto a lo que se tiene que hacer y no cómo se va a hacer. En esta sección se modelan las clases del análisis. Se ofrece una especificación más precisa de los requisitos que la que se obtuvo como resultado de la captura de requisitos. Se analizan y refinan los requerimientos que fueron descritos en el diagrama de casos de uso para alcanzar una visión detallada de los mismos; mostrando una vista interna del sistema, estructurado por clases estereotipadas (ver tabla 3). A continuación se explican los estereotipos presentes en los diagramas de clases del análisis:

**Tabla 3. Estereotipos del diagrama de análisis**

Clase	Estereotipos	Descripción
Entidad		Su función es modelar información que posee larga vida y que es a menudo persistente.
Interfaz		Es la encargada de modelar la interacción entre el sistema y sus actores.
Control		Su función es regular la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

### 3.2.1 Diagramas de clases del análisis

A continuación se muestra el diagrama de clases del análisis<sup>13</sup> del caso de uso **Reproducir texto**. Para ver los diagramas de los restantes casos de uso remitirse al **Anexo # 5: Diagramas de clases del análisis**.



**Figura 4. Diagrama de clases del análisis del CU Reproducir texto**

<sup>13</sup> Una clase de análisis representa una abstracción de una o varias clases y / o subsistemas del diseño del sistema.

### 3.2.2 Diagramas de colaboración

Un diagrama de colaboración modela las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a esos enlaces. Se representa una combinación de la información prevista por el diagrama de casos de uso y el diagrama de clases del análisis describiendo tanto la estructura estática como el comportamiento dinámico de la funcionalidad que abarca. A continuación se muestra el diagrama de colaboración del caso de uso **Reproducir texto. Sección reproducir todo el contenido del sitio web**. Para el estudio de los restantes diagramas remitirse al **Anexo # 6: Diagramas de colaboración**.

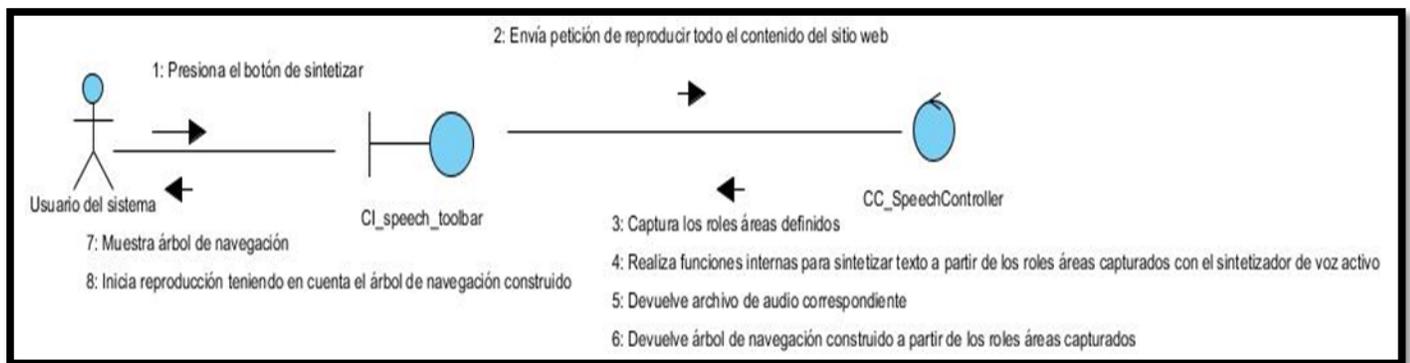


Figura 5. Diagrama de colaboración del CU Reproducir texto. Sección reproducir todo el contenido del sitio web

### 3.3 Patrón arquitectónico Modelo Vista Controlador (en lo adelante MVC) en Symfony

El Sistema propuesto se desarrollará como un *bundle* de Symfony 2.3.4, por ello comparte una arquitectura similar a otros componentes desarrollados para este framework. No obstante, posee características propias que lo distinguen de soluciones anteriores. Utiliza MVC, pero tiene su propia forma de trabajo, con variantes del mismo como la capa de abstracción de base de datos, el controlador frontal y las acciones. Según (Eguiluz, 2012), este patrón separa el código en tres capas, los datos (*model*), la interfaz de usuario (*view*) y la lógica de negocio (*controller*). Esto permite una mayor reutilización de componentes, facilita las modificaciones del aspecto del programa/web sin tener que tocar el funcionamiento y simplifica el mantenimiento. Funciona de forma “triangular”, es decir, cada capa envía información a una y la recibe de la otra.

A continuación se mostrará la descripción de la aplicación del patrón MVC en Symfony 2, dada por (Eguiluz, 2012)

El modelo es la representación de la información y se encarga de gestionar el acceso, adición y modificación de los mismos (operaciones CRUD). A nivel de interacción con los otros componentes envía a la vista los datos que tienen que ser mostrados y recibe desde el controlador las peticiones de acceso y manipulación.

Cuando el usuario solicita ver la portada del sitio, internamente sucede lo siguiente:

1. El sistema de enrutamiento determina qué Controlador está asociado con la página de la portada.
2. Symfony2 ejecuta el Controlador asociado a la portada. Un controlador no es más que una clase PHP en la que puedes ejecutar cualquier código que quieras.
3. El Controlador solicita al Modelo los datos asociados. El modelo no es más que una clase PHP especializada en obtener información, normalmente de una base de datos.
4. Con los datos devueltos por el Modelo, el Controlador solicita a la Vista que cree una página mediante una plantilla y que inserte los datos del Modelo.
5. El Controlador entrega al servidor la página creada por la Vista.

En general el funcionamiento interno de symfony 2, siempre es el mismo: 1) el Controlador manda y ordena, 2) el Modelo busca la información que se le pide, 3) la Vista crea páginas con plantillas y datos.

### **3.4 Patrones de diseño**

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. Brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Entre las principales ventajas de un framework, se evidencia la habilidad de simplificar el desarrollo de aplicaciones mediante la automatización de algunos de los

patrones utilizados para resolver las tareas comunes, característica que hace posible la gran usabilidad que tienen, casi siempre independientes del tipo de aplicación web que se desea desarrollar. El framework Symfony, evidencia el uso de varios patrones de diseño que incluye por defecto en su arquitectura, estando concebido en función de obligar al programador a aplicarlos como por ejemplo los patrones GRASP (Patrones Generales de Software para Asignación de Responsabilidades, traducido al inglés *General Responsibility Assignment Software Patterns*) y GoF (Grupo de 4, traducido al inglés *Gang of Four*). (Tedeschi, 2014)

### **Patrones GRASP**

- ✓ **Experto:** Symfony lo utiliza en la inclusión de la librería Doctrine para mapear la base de datos. (Tedeschi, 2014)
- ✓ **Creador:** es utilizado por el sistema, fundamentalmente en las clases controladoras cuando se crean objetos de las clases que representan las entidades, o se instancian las mismas. (Tedeschi, 2014)
- ✓ **Bajo acoplamiento:** las clases controladoras (*Controllers*) implementadas en el componente heredan solamente de la clase *Controller* del framework, lo mismo ocurre con las clases relacionadas con el modelo y la lógica del negocio. El bajo acoplamiento también se evidencia en otras partes del componente, donde se evita que las clases tengan demasiadas dependencias. (Tedeschi, 2014)
- ✓ **Alta cohesión:** la información que almacena cada una de las clases debe estar relacionada con la clase y ser coherente. Un ejemplo de alta cohesión es la clase controladora del componente que es capaz de ejecutar tareas interactuando muy poco o nada con otros componentes del framework. (Tedeschi, 2014)
- ✓ **Controlador:** está incluido en el patrón MVC que implementa Symfony, a través del cual todas las peticiones web son interceptadas por un único punto de entrada o controlador frontal (*app.php* o *app\_dev.php*) y procesadas posteriormente por las clases apropiadas, en este caso los controladores (*Controllers*). (Tedeschi, 2014)

### **Inyección de Dependencias (DI en lo adelante por sus siglas en inglés)**

Es una herramienta comúnmente utilizada en varios patrones de diseño orientado a objetos, consiste en inyectar comportamientos a componentes. Consiste en extraer responsabilidades a un componente para delegarlas en otro, estableciendo un mecanismo a través del cual el nuevo componente pueda ser cambiado en tiempo de ejecución. (Pacheco, 2014)

La implementación habitual en programación es crear un método capaz de establecer el comportamiento, es decir capaz de cambiar el valor de un atributo asignándole una instancia de objeto diferente. La aplicación de este patrón permite la adición de nuevos sintetizadores al componente sin tener que realizar cambios al mismo, basta con que un desarrollador implemente algunas clases e interfaces definidas y configure su sintetizador. (Pacheco, 2014)

### **Patrones GoF**

- ✓ **Estrategia:** se encuentra dentro de los patrones de comportamiento y permite disponer de varios procedimientos para realizar un CU y elegir cuál es el que se va a aplicar en tiempo de ejecución. Con él se pueden definir un conjunto de algoritmos para un mismo método; pudiendo seleccionar a cada instante cómo debe comportarse la clase cliente. (Visconti, et al., 2014) (Sánchez, 2008)

### **3.5 Modelo del diseño**

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en las funcionalidades que se desean implementar.

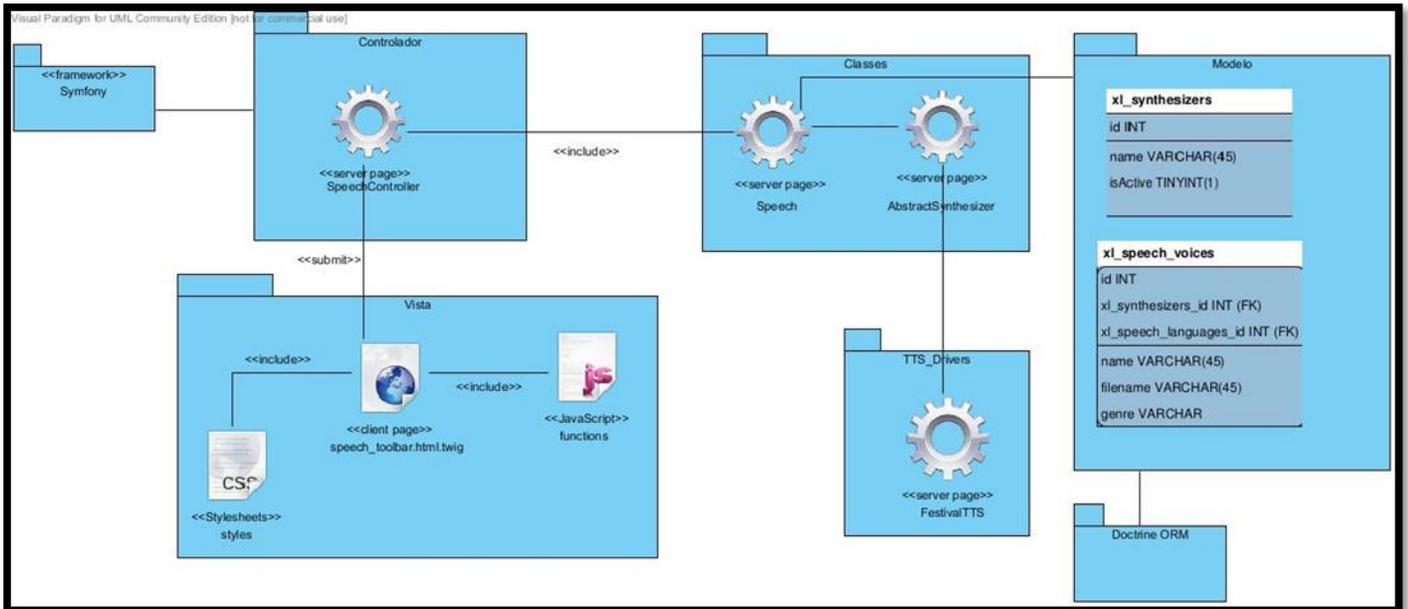
### **3.6 Diagrama de clases del diseño**

A continuación se describen los elementos principales por los que están compuestos los diagramas de clases del diseño; las diferencias entre estos radican en las particularidades que cada CU contiene. Es

necesario precisar que con el objetivo de agrupar las clases según el patrón MVC que Symfony propone se han creado tres paquetes para almacenar las clases denominados: Modelo, Vista y Controlador.

- ✓ **Modelo:** En este paquete se encuentran las clases que se encargan de la abstracción y acceso a los datos almacenados en la base de datos. Estas clases son generadas por el ORM Doctrine y permiten manipular las tablas de la base de datos como objetos.
- ✓ **Vista:** El componente propuesto está diseñado para visualizarse dentro del layout de aplicaciones web que requieran un lector de pantalla. Debido a esto, no se define un layout general, sino una barra de herramientas (*speech\_toolbar.html.twig*) con los elementos requeridos para que el usuario lo utilice. También se define una vista para acceder a la configuración de los parámetros del componente (*configuration.html.twig*).
- ✓ **Controlador:** Las clases controladoras se encargan de atender las peticiones y visualizar las páginas correspondientes. Se crearon 2 clases controladoras (*SpeechController.php*), que se encarga de trabajar con una herramienta externa para sintetizar textos y (*ConfigController.php*) para la configuración general del componente.
- ✓ **Framework Symfony:** Contiene todos los componentes del framework utilizado. Dentro de este se incluyen: Doctrine ORM para la conexión y manejo con la base de datos, las clases para los controladores, el sistema de enrutamiento, el sistema de plantillas Twig para las vistas.
- ✓ **Componente Synthesizer:** representa una aplicación externa, la cual es un ejecutable en el sistema a través del que se realizará la síntesis de voz. De acuerdo a la configuración del componente pudiera ser alguno de los sintetizadores más conocidos como FestivalTTS, Loquendo, Jaws o Soldescott.

Seguidamente se presenta el diagrama de clases del diseño correspondiente al caso de uso **Reproducir texto**. Para ver los diagramas correspondientes, remitirse al **Anexo # 7 Diagramas de diseño de clases de los casos de uso**.



**Figura 6. Diagrama de clases de diseño del CU Capturar texto**

### 3.7 Diagrama de Despliegue

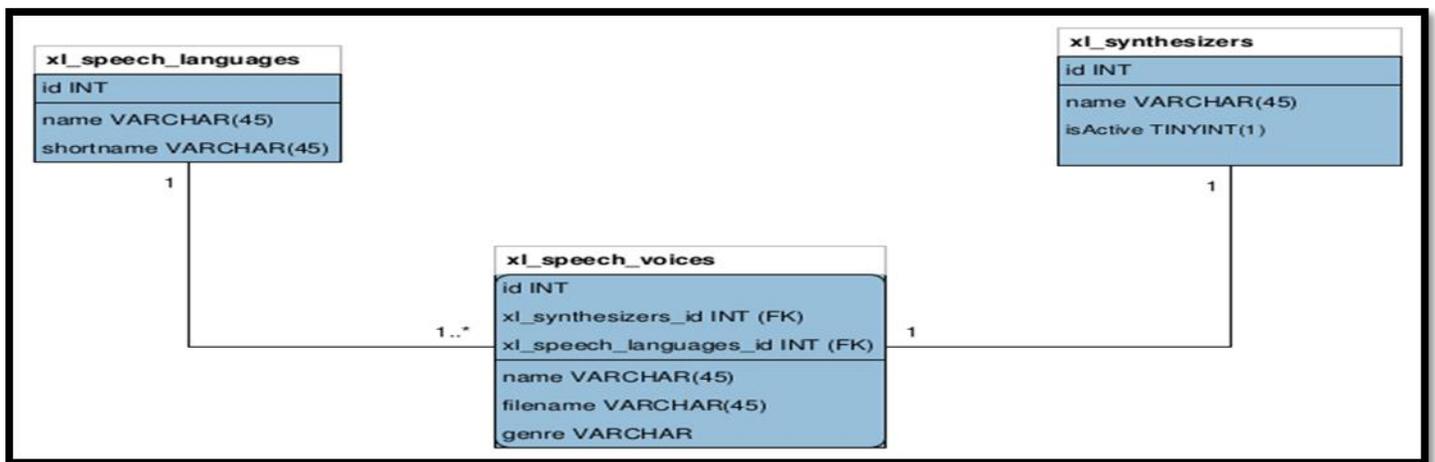
El diagrama de despliegue que se muestra a continuación representa la distribución física del sistema a través de nodos. Está compuesto por una PC Cliente que deberá tener instalado un navegador Web, donde la comunicación entre ella y el servidor se llevará a cabo a través del protocolo HTTP, en este servidor estará instalado el sistema con el sintetizador de voz configurado y su respectiva base de datos.



**Figura 7. Diagrama de despliegue**

### 3.8 Diseño de la base de datos

Un diagrama de base de datos está orientado a explicar el conjunto de tablas con las que interactúa el sistema para garantizar su funcionalidad. En este caso el componente propuesto se centra en la interacción con una herramienta externa para efectuar la síntesis de texto a voz. A continuación se presenta el modelo de datos que contiene las entidades que serán utilizadas por las funcionalidades a desarrollar y las relaciones entre ellas. Estas entidades representan las tablas en la base de datos en las que se almacenan (o almacenarán) todos los datos relacionados con los sintetizadores.



**Figura 8. Diagrama de diseño de la base de datos**

### 3.9 Descripción de las tablas de la base de datos

**Tabla 4. Descripción de la tabla xl\_speech\_languages de la base de datos**

XL_speech_languages		
<b>Descripción:</b> Esta tabla almacena los lenguajes disponibles por cada sintetizador.		
Atributo	Tipo	Descripción
Id	INT	Identificador único de la tabla.

name	VARCHART (45)	Se muestra el nombre de las voces y los idiomas al usuario, en dependencia de los que soporte.
shortname	VARCHART (45)	Es una abreviatura de los nombres de las voces disponibles de acuerdo con las soportadas por el sintetizador.

**Tabla 5. Descripción de la tabla xl\_synthesizers de la base de datos**

XL_synthesizers		
<b>Descripción:</b> Esta tabla almacena los sintetizadores disponibles para el sistema y además sus respectivos valores.		
Atributo	Tipo	Descripción
Id	INT	Identificador único de la tabla.
name	VARCHART (45)	Se muestra el nombre de los sintetizadores disponibles en el componente.
isActive	TINYINT	Representa si el sintetizador está activo o no.

**Tabla 6. Descripción de la tabla XL\_speech\_voices de la base de datos**

XL_speech_voices		
<b>Descripción:</b> Esta tabla almacena las voces disponibles para cada sintetizador del sistema.		
Atributo	Tipo	Descripción
Id	INT	Identificador único de la tabla.
name	VARCHART (45)	Se almacena el nombre de las voces disponibles para cada uno de los sintetizadores del componente.
filename	VARCHART (45)	Se muestran al usuario un nombre entendible, que ha sido asignado al sintetizador para ser mostrado por el

		sistema.
genre	VARCHART	Se muestra al usuario el género (masculino o femenino) de las voces de los sintetizadores.

### 3.10 Conclusiones

Los diagramas de clases del análisis y del diseño que se obtuvieron, han sido dirigidos a dar cumplimiento a los distintos requisitos funcionales y no funcionales seleccionados. Se logró obtener de forma concreta y detallada las relaciones entre clases, además de un diseño claro de la base de datos. Se definieron correctamente los patrones arquitectónicos y patrones de diseño a emplear durante el desarrollo del componente. Se realizó el diagrama de despliegue que permitió obtener los nodos físicos y las asociaciones de comunicación existentes.

# Capítulo # 4 Implementación y prueba de las funcionalidades del componente

## 4.1 Introducción

En este capítulo se presentan los artefactos generados en el diseño; los cuales son la entrada fundamental al flujo de trabajo de **implementación** y tiene como objetivo, mostrar cómo se implementan las funcionalidades en términos de componentes y cómo estos se organizan. Luego se realizan las **pruebas** necesarias a las funcionalidades permitiendo verificar cuan correcto es su desarrollo.

## 4.2 Modelo de implementación

Esta sección muestra el Modelo de implementación, conformado por los componentes, subsistemas y módulos que representan la composición física de la implementación del componente final. Se describe cómo es que los elementos del modelo de diseño se implementan en términos de componentes físicos (datos, archivos, ejecutables, código fuente y los directorios) y módulos, relacionando entre si los paquetes y clases obtenidos, agrupándolos en una estructura organizada disponibles en el entorno de implementación y en el lenguaje de programación utilizado y señalando las dependencias que existen entre estos. En el modelo de implementación se obtienen los diagramas de componentes que a continuación se presentan.

### 4.2.1 Diagramas de componentes

Los diagramas de componentes se utilizan para modelar la implementación estática del sistema. Es un tipo especial de diagrama de clases que se centra en los componentes de un sistema, su organización y las dependencias entre estos; siendo componentes todos aquellos elementos que forman parte de la solución brindada a las funcionalidades del sistema. Los componentes físicos incluyen archivos, cabeceras, librerías compartidas, módulos, ejecutables o paquetes. Pueden ser usados para modelar y

documentar cualquier arquitectura de sistema. Debido a que estos son muy parecidos a los diagramas de casos de usos son utilizados para modelar la vista estática de un sistema.

A continuación se muestra el diagrama de componentes del caso de uso **Reproducir texto** y la descripción de los principales elementos que lo conforman. Para el estudio de los demás diagramas remitirse al **Anexo # 8: Diagramas de componentes**.

- ✓ **Paquete Vista:** Paquete que contiene componentes pertenecientes a la vista del CU.
- ✓ **Paquete Controlador:** Paquete que agrupa los componentes que representan las acciones del CU.
- ✓ **Paquete Modelo:** Paquete que agrupa las clases para acceder a los datos de la base de datos, incluyendo las generadas por el ORM Doctrine y las creadas por los desarrolladores.
- ✓ **Paquete TTS\_Drivers:** Paquete que agrupa las clases que heredan de ***AbstractSynthesizer.php***, las cuales representan los sintetizadores de voz disponibles en el sistema (**Ejemplo *FestivalTTS.php, EspeakTTS.php***).
- ✓ **Componente Base de datos:** Encapsula todos los datos del sistema.
- ✓ **Componente CSS:** Archivos que se encargan de la visualización del contenido de las plantillas.
- ✓ **Componente Symfony:** Agrupa los elementos del framework que intervienen en la realización del CU.
- ✓ **Componente config:** Contiene los archivos donde se almacena la configuración del componente en sus diferentes comportamientos.
- ✓ **Componente lib:** Almacena las bibliotecas y clases del componente y del sistema en general.
- ✓ **Componente jQuery:** Contiene los archivos con código JavaScript que se ejecutan en la realización del CU y mantienen el dinamismo de la página web.

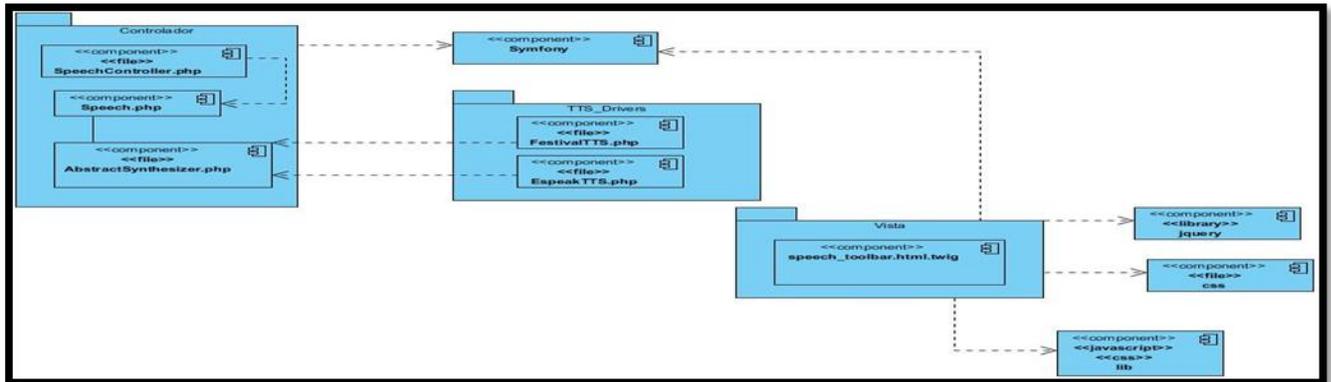


Figura 9. Diagrama de componentes del CU Reproducir texto

### 4.3 Pruebas de software

En esta sección se presentan las pruebas funcionales realizadas al componente con el objetivo de detectar los posibles fallos en la aplicación.

#### 4.3.1 Niveles de pruebas

##### Pruebas unitarias:

Una prueba unitaria normalmente es una prueba contra una clase *PHP* específica. En *Symfony 2*, las pruebas unitarias son muy parecidas a las de *PHPUnit* normales. Las pruebas unitarias se encargan de garantizar que unidades individuales de código funcionan correctamente cuando se utilizan de manera aislada. En un código base orientado a objetos tal como lo es *Symfony2*, una unidad sería una clase y sus métodos. (Eguiluz, 2012)

Al escribir las pruebas unitarias, los casos de prueba se deben escribir de forma independiente de otros casos de prueba. Cuando se prueban unidades, es útil poder crear objetos simulados que permitan probar fácilmente las funciones de unidades que tienen dependencias externas. La imitación permite simular una llamada a función en lugar de ejecutarla realmente. Las pruebas unitarias no prueban que los componentes de una aplicación funcionan correctamente en conjunto, sino que su código esté escrito correctamente, en la forma más optimizada posible. (Eguiluz, 2012)

**Pruebas de integración:**

La prueba de integración se realiza posteriormente a las pruebas de unidad y su foco de atención es el diseño y la construcción de la arquitectura del software. Después de la integración viene la validación y por último la prueba del sistema. (Rhapsody, 2014)

**Pruebas de validación:**

La validación se logra cuando el software funciona de acuerdo a las expectativas del cliente. Se consigue mediante una serie de pruebas de la caja negra que demuestran la conformidad con los requisitos. Si aparecen deficiencias, hay que negociar con el cliente un método para resolverlas. (Rhapsody, 2014)

La prueba alfa es conducida por un cliente en el lugar de desarrollo y se llevan a cabo en un entorno controlado. Se usa el software de manera natural, con el encargado de desarrollo "mirando por encima del hombro" del usuario" y registrando errores y problemas de uso. (Rhapsody, 2014)

La prueba beta se lleva a cabo en uno o más lugares de clientes por los usuarios finales del software. A diferencia de la prueba alfa, el encargado de desarrollo, normalmente, no está presente. La prueba beta es una aplicación "en vivo" del software en un entorno que no puede ser controlado por el equipo de desarrollo. El cliente registra todos los problemas (reales o imaginarios) que encuentra y los informa a intervalos regulares. Como resultado, el equipo de desarrollo lleva a cabo modificaciones y así prepara una versión del producto para toda la base de clientes. (Rhapsody, 2014)

**Pruebas de sistema:**

El objetivo de estas pruebas es probar a fondo el sistema, comprobando su funcionalidad e integridad globalmente, en un entorno lo más parecido posible al entorno final de producción. Estas, permiten sistematizar el proceso, validar de forma temprana los requisitos para corregir las deficiencias que presenten y asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación. (Pons, et al., 2010)

### 4.3.2 Métodos de prueba

Las técnicas de evaluación dinámica o prueba proveen diferentes criterios para generar casos de prueba, en lo adelante CP, que buscan los fallos que pueda tener el componente. Estas técnicas se agrupan en: pruebas de Caja negra y Caja blanca, las cuales se explican a continuación.

#### **Pruebas de caja negra:**

Las pruebas de caja negra o pruebas de comportamiento se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. El componente se ve como una “Caja Negra” cuyo comportamiento sólo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas. Para seleccionar el conjunto de entradas y salidas sobre las que trabajar, hay que tener en cuenta que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo en nuestro sistema, y como consecuencia producen una serie de salidas que revelan la presencia de defectos. Como la prueba exhaustiva es imposible, el objetivo final es, encontrar una serie de datos de entrada cuya probabilidad de pertenecer al conjunto de entradas que causan dicho comportamiento erróneo sea lo más alto posible. Al igual que ocurría con las técnicas de Caja Blanca, para confeccionar los casos de prueba de Caja Negra existen distintos criterios. Algunos de ellos son: (Angel 2008)

- ✓ **Particiones de Equivalencia:** dividen el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Esto quiere decir que si el caso de prueba correspondiente a una clase de equivalencia detecta un error, el resto de los casos de prueba de dicha clase de equivalencia deben detectar el mismo error y viceversa. El diseño de casos de prueba según esta técnica consta de dos pasos: (Angel 2008)
  - ✓ Identificar las clases de equivalencia. (Angel 2008)
  - ✓ Identificar los casos de prueba. (Angel 2008)
- ✓ **Análisis de Valores Límite:** las pruebas realizadas bajo estas condiciones, producen mejor resultado que aquellos que no lo hacen. Las condiciones límite son aquellas que se hayan en los márgenes de la clase de equivalencia, tanto de entrada como de salida. (Angel 2008)

El análisis de valores límite complementa la técnica de partición de equivalencia de manera que:

- ✓ En lugar de seleccionar cualquier caso de prueba de las clases válidas e inválidas, se eligen los casos de prueba en los extremos. (Angel 2008)
- ✓ En lugar de centrarse sólo en el dominio de entrada, los casos de prueba se diseñan también considerando el dominio de salida. (Angel 2008)

A continuación se puntualiza la principal prueba de sistema realizada al componente con el uso del método de Caja negra:

- ✓ **Adivinando el error:** Durante el desarrollo del componente, por la intuición y la experiencia, se evidenciaron ciertos tipos probables de errores. A partir de los CU y teniendo en cuenta dichos errores se describieron los CP, de manera tal que pudieran ser detectadas no conformidades relacionadas a los mismos.

#### **Pruebas de caja blanca:**

En programación, se denominan cajas blancas a un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo. Las pruebas de caja blanca están dirigidas a las funciones internas. Entre las técnicas usadas se encuentran; la cobertura de caminos (pruebas que hagan que se recorran todos los posibles caminos de ejecución), pruebas sobre las expresiones lógico-aritméticas, pruebas de camino de datos (definición-uso de variables), comprobación de bucles (se verifican los bucles para 0,1 y n iteraciones, y luego para las iteraciones máximas, máximas menos uno y más uno. (Angel 2008)

Las pruebas de caja blanca se llevan a cabo en primer lugar, sobre un módulo concreto, para luego realizar las de caja negra sobre varios subsistemas (integración). (Angel 2008)

En la misma medida en que la funcionalidades del componente fueron siendo desarrolladas, su correcta implementación fue siendo comprobada a través de pruebas unitarias. Estas fueron realizadas por la desarrolladora utilizando el método de Caja blanca. No se planificaron ni se registraron sus resultados ya que se llevaron a cabo durante todo el proceso de implementación.

### 4.3.3 Diseño de Casos de Pruebas

En los CP derivados, se incluyen los datos de entrada y los resultados esperados. A continuación se presenta el CP **Reproducir texto**; para consultar el resto remitirse al **Anexo # 9: Diseños de casos de prueba**.

**Tabla 7. Descripción del CP Reproducir texto**

CP Reproducir texto							
<b>Descripción general</b>							
El caso de uso inicia cuando el actor del sistema selecciona un texto para reproducirlo. Una vez reproducido el texto, finaliza el caso de uso.							
<b>Condiciones de ejecución</b>							
El usuario accede a un sitio web que tenga instalado el componente.							
El usuario define el texto a reproducir.							
SC1 Reproducir texto							
Escenario	Descripción	Etiquetas de role.	Textos alternativos de las imágenes	Información relevante del sitio web	Respuesta del sistema	Flujo central	
EC 1.1	Selecciona texto que se desea reproducir.	N/A	N/A	N/A	Captura el texto seleccionado y permite reproducirlo.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.	

**CAPÍTULO 4 | UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS | FACULTAD 4**  
**COMPONENTE PARA LA INTEGRACIÓN DE UN LECTOR DE PANTALLA AL MARCO DE TRABAJO XALIX**

	No selecciona texto a reproducir.	V	V	V	Permite reproducir todo el contenido del sitio web.	
		V	V	I	Captura los datos existentes asociados a las variables con valores válidos y permite reproducirlos.	
		V	I	V		
		I	V	V		
		I	V	I		
		I	I	V		
		I	I	I		
EC 1.2	Se procesa el texto, normalizándolo y sintetizándolo para convertirlo en un archivo de audio que podrá ser interpretado y reproducido dentro de un navegador web.	N/A	N/A	N/A	Se prepara para iniciar la reproducción del texto.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 1.3	Selecciona la opción reproducir texto	N/A	N/A	N/A	Inicia la reproducción del texto.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.

**SC2 Iniciar reproducción mediante atajos del teclado**

**CAPÍTULO 4 | UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS | FACULTAD 4**  
**COMPONENTE PARA LA INTEGRACIÓN DE UN LECTOR DE PANTALLA AL MARCO DE TRABAJO XALIX**

EC 2.1	Presiona la combinación de teclas <b>Ctrl+</b> ,	N/A	N/A	N/A	Si existe algún texto previamente seleccionado, el sistema procede a su reproducción en otro caso reproduce todo el contenido del sitio web.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
<b>SC3 Detener reproducción del texto</b>						
EC 3.1	Presiona la tecla espaciadora.	N/A	N/A	N/A	El sistema detiene la reproducción del texto.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
<b>SC4 Adelantar reproducción del texto</b>						
EC 4.1	Presiona la tecla Ctrl+*	N/A	N/A	N/A	El sistema adelanta la reproducción del texto, al bloque siguiente definido por los roles de WAI-ARIA.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.

EC 4.2	Presiona la tecla Ctrl+/ 	N/A	N/A	N/A	El sistema retrocede la reproducción del texto, al bloque anterior definido por los roles de WAI-ARIA.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 4.3	Utiliza el <i>mouse</i> para marcar el tiempo de la reproducción al cuál desea ir.	N/A	N/A	N/A	El sistema inicia reproducción de texto a partir del tiempo definido.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.

#### 4.3.4 Resultados obtenidos

Con el objetivo de detectar la mayor cantidad de no conformidades posibles en las funcionalidades del componente se realizaron varias iteraciones de prueba las cuales son descritas a continuación:

**Primera iteración:** Fueron realizadas por 4 estudiantes de 5to año, de distintos proyectos, guiados por los CP y dirigidas por la desarrolladora del componente. De esta forma los estudiantes fueron observando y registrando los errores y problemas de uso encontrados.

A continuación se muestra una relación de las no conformidades encontradas durante la iteración:

#### CU Reproducir texto:

1. La reproducción de texto a través de atajos del teclado no funciona.
2. No se reproducen los textos alternativos de las imágenes.

#### CU Reproducir texto. Sección reproducir todo el contenido del sitio web:

1. Esta funcionalidad no está disponible.

**CU Descargar archivo de audio:**

1. No se descarga el archivo de audio de todo el sitio web.

**Tabla 8. Resultados de la primera iteración de pruebas**

Casos de Prueba	No conformidades				
	Alta	Media	Baja	No procede	Total
Reproducir texto	-	-	2	-	2
Reproducir todo el contenido del sitio web	1	-	-	-	1
Adelantar reproducción	-	-	-	-	-
Detener reproducción	-	-	-	-	-
Descargar archivo de audio	-	1	-	-	1

**Segunda iteración:** La segunda iteración de pruebas fue realizada por un equipo compuesto por 4 trabajadores del departamento de las TIC, en la universidad de Ciencias Médicas de Ciego de Ávila. El objetivo perseguido era comprobar que las no conformidades detectadas anteriormente fueron solucionadas.

Las pruebas internas realizadas permitieron detectar 4 no conformidades. De ellas 1 de complejidad alta, 1 de complejidad media y 2 de complejidad bajas.

En su mayoría estas no conformidades han sido de funcionalidades del componente. Se fueron solucionando en la medida en que fueron detectadas y han servido para garantizar que en otras iteraciones no se repitan.

#### **4.4 Conclusiones**

Al finalizar el capítulo, quedó realizado el modelo de implementación, donde se modelaron los diagramas de componentes que ilustran las relaciones de los componentes físicos donde se encuentran implementadas las funcionalidades y sus dependencias con la base de datos. Se describieron los CP que se emplearon en la realización de las pruebas al componente. Esta etapa del desarrollo se caracterizó por obtener resultados visibles tanto para los que se benefician con la solución brindada como para el equipo de trabajo, ya que quedan completamente implementadas las funcionalidades que se definieron al inicio de la investigación. Es por ello que se puede afirmar que los resultados fueron satisfactorios, puesto que se identificaron a tiempo los problemas presentados por la aplicación.

## Conclusiones

Al finalizar la investigación realizada se puede afirmar que se cumplieron los objetivos trazados, obteniéndose como principal el componente, además se pudo concluir lo siguiente:

- ✓ El estudio del arte realizado sobre las herramientas similares, apoyó la selección de las mejores prácticas durante el desarrollo del componente y sirvió de base para la posterior implementación de las funcionalidades, permitiendo con esto seleccionar los mejores elementos presentes en estos sistemas y aplicarlos al desarrollo del componente.
- ✓ El componente desarrollado favorecerá la accesibilidad a la web, por parte de cualquier usuario y realizará aportes al contenido presente en las plataformas educativas; ya que los lectores de pantalla son herramientas capaces de facilitar esto.
- ✓ La tecnología WAI-ARIA, es de gran importancia en el proceso de lectura de pantalla, pues los roles área que define, facilitan la navegación por sitios web y la reproducción de todo su contenido.
- ✓ Los sintetizadores de texto son el núcleo fundamental de los lectores de pantalla, pues son la herramienta capaz de traducir la representación simbólica en audios, favoreciendo el proceso de la lectura de pantalla.

## Recomendaciones

Teniendo en cuenta los resultados obtenidos en esta investigación y basado en la experiencia adquirida, se recomienda:

- ✓ Se recomienda al equipo de desarrollo de Xalix, que incorporen al estándar de codificación la inclusión de los roles áreas definidos por la tecnología WAI-ARIA en el sitio de la [www .w3.org](http://www.w3.org).
- ✓ Se recomienda al equipo de desarrollo de Xalix que se extienda la capacidad de leer todo el contenido del texto a través de los roles áreas definidos por la tecnología WAI-ARIA, ya que hasta el momento solo se interpretan los siguientes roles: *banner*, *navigation*, *link* y *main*.
- ✓ Se recomienda al departamento de Vigilancia Tecnológica el estudio de los sintetizadores de voz, y las voces que estos pueden soportar.
- ✓ Se recomienda a la UCI, la adquisición de voces como las de *Loquendo*, que aunque no sean gratuitas cuentan con una gran naturalidad e inteligibilidad.

## Referencias bibliográficas

**ADDKW. 2012.** PMI una definición y una aplicación real al desarrollo de un Sistema Logístico. [En línea] 08 de junio de 2012. <http://addkw.com/2012/06/08/pmi-una-definicion-y-una-aplicion-real-al-desarrollo-de-un-sistema-logistico/>.

**Alonso, Méndez, Evelyn. 2009.** Herramientas CASE para el proceso de desarrollo de Software. [En línea] 2009. <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.

**Álvarez, Ángel, Miguel. 2003.** Zend Studio. [En línea] 04 de junio de 2003. <http://www.desarrolloweb.com/articulos/1178.php>.

**Arnet, Alain, Nestor. 2014.** Informática. [En línea] 2014. [http://informatica.cubaeduca.cu/index.php?option=com\\_content&view=article&id=11212:modelo-relacional-teoria-de-normalizacion-1fn&catid=432&Itemid=184](http://informatica.cubaeduca.cu/index.php?option=com_content&view=article&id=11212:modelo-relacional-teoria-de-normalizacion-1fn&catid=432&Itemid=184).

**AT&T Labs. 2011.** Text-to-Speech. [En línea] 2011. <http://www2.research.att.com/~ttsweb/tts/index.php#top>.

**Broche, Salvador, Felipe, Orlando. 2014.** 2014.

**Carrasco, Arlet, Puebla, Ing.Yoan y Gómez, Chaviano Ing. Enrique. 2009.** Arquitectura orientada a servicios para el módulo de inventario del ERP cubano. [En línea] 2009. <http://www.gestiopolis.com/administracion-estrategia/erp-arquitectura-orientada-a-servicios.htm>.

**Chavez, Alan. 2013.** Symfony: El mejor framework de PHP. 2013.

**CJava. 2013.** Disponible PostgreSQL 9.3. [En línea] 11 de septiembre de 2013. <http://cjavaperu.com/blog/?p=1528>.

**Díaz, Pereda, Vladimir, Ing. Ernesto y Pons, Miranda, Ing. Yaismel. 2013.** *Pautas de Codificación*. Habana : s.n., 2013.

**Diccionario de informática. 2014.** Definición de UML. [En línea] 2014. <http://www.alegsa.com.ar/Dic/uml.php>.

**Eberlei, Benjamin. 2012.** Doctrine 2.3 final relased — Doctrine Project. [En línea] 20 de septiembre de 2012. <http://www.doctrine-project.org/2012/09/20/doctrine-2-3-final.html>.

**Eguiluz, Javier. 2012.** *Desarrollo web ágil con Symfony 2*. s.l. : easybook v 4.8-DEV, 2012.

**EIAlecs. 2006.** TTS (Text To Speech ) usando Festival y MBrola en Ubuntu. [En línea] 2006. <http://huevon.blogspot.com/2006/01/tts-text-to-speech-usando-festival-y.html>.

- Fernández, Varona, Amparo. 1997.** *Antecedentes y desarrollo de los sistemas actuales de reconocimiento automático del habla*. s.l. : CICYT, 1997. ISBN-TIC95-0884-C04-03.
- FireVox. 2014.** Fire Vox: A Screen Reading Extension for Firefox. [En línea] enero de 2014. Fire Vox: A Screen Reading Extension for Firefox.
- Flores, Díaz, Milagro, Mirian. 2014.** Rational Unified Process vs Extreme Programming. 2014.
- García, Fernando. 2013.** Entorno de Desarrollo Integrado (IDE). [En línea] 2013. <http://fergarcia.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>.
- Gómez, Landeros, Priscila, Ruth. 2014.** Herramientas Case. [En línea] 2014. <http://www.monografias.com/trabajos14/herramicase/herramicase.shtml>.
- Gregorio, José. 2011.** Servidor web. [En línea] 28 de junio de 2011. <http://www.slideshare.net/josegregoriob/servidor-web-8451426>.
- GW Micro. 2014.** Lectores de pantalla. [En línea] 2014. <http://www.gwmicro.com/Window-Eyes/>.
- Jacobson, Ivan, Booch, Grady y Rumbaugh, James. 2000.** *El proceso unificado de desarrollo de software*. Madrid : Pearson Educación. S.A., 2000. ISBN-84-7829-036-2.
- Lighttpd. 2014.** Lighttpd - fly light. [En línea] 2014. <http://www.lighttpd.net/>.
- Luján, Mora, Sergio. 2007.** ¿Qué es un lector de pantalla? [En línea] 18 de Marzo de 2007. <http://accesibilidadenlaweb.blogspot.com/2007/03/qu-es-un-lector-de-pantalla.html>.
- Luna, Castelazo, Yanin. 2007.** *Desarrollo de una aplicación de síntesis de voz con características prosódicas*. Unidad Iztapalapa : Universidad autónoma metropolitana, 2007. 200216938.
- masadelante.com. 2014.** ¿Qué es un plug-in? [En línea] 2014. <http://www.masadelante.com/faqs/plug-in>.
- MundoAccesible. 2009.** ESpeak - Por un Mundo Accesible. [En línea] 01 de julio de 2009. <http://wiki.mundoaccesible.org.ve/index.php?title=ESpeak>.
- MurdenManX. 2010.** Pmbok ¿Qué es? [En línea] 2010. <http://www.buenastareas.com/ensayos/Pmbok-Qu%C3%A9-es/128400.html>.
- Netbeans. 2014.** Netbeans. [En línea] 2014. [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html).
- Olabe, Basogain, Xavier. 2013.** Redes Neuronales y sus aplicaciones. 2013.
- Oviedo, Carlos. 2013.** La historia del sintetizador. [En línea] 22 de noviembre de 2013. <http://www.taringa.net/posts/info/3510820/La-historia-del-sintetizador.html>.

- Pacheco, Nacho. 2011.** Doctrine2ORM. [En línea] 3 de Noviembre de 2011. [http://sf2-es.net16.net/\\_downloads/Doctrine2ORM.pdf](http://sf2-es.net16.net/_downloads/Doctrine2ORM.pdf).
- Pacheco, Ruiz, Carlos, Juan. 2014.** [En línea] 12 de marzo de 2014. <http://msdn.microsoft.com/es-es/library/jj635998.aspx>.
- Pachersky, Verónica. 2012.** Normalización de un texto de entrada para un sistema de síntesis del habla. Buenos Aires : Universidad de Buenos Aires, 2012.
- Pérez, Eguiluz, Javier. 2013.** Desarrollo web ágil con Symfony 2. 2013.
- Pons, Claudia, Giandini, Roxana y Pérez, Gabriela. 2010.** *Desarrollo de software dirigido por modelos. Conceptos teóricos y su aplicación práctica.* Buenos Aires : edulp Editorial de la Universidad de la Plata, 2010. ISBN 978-950-34-0630-4.
- PowerData. 2013.** ¿Qué son los procesos ETL? [En línea] 06 de junio de 2013. <http://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/288844/Qué-son-los-procesos-ETL>.
- Pressman, S. Roger. 1998.** *Ingeniería de Software. Un enfoque práctico.* Madrid : McGraw-Hill, 1998. ISBN-8448111869.
- Reverso. 2005.** Traducción anagrafico Inglés | Diccionario Italian | Reverso. [En línea] 2005. <http://diccionario.reverso.net/italian-ingles/anagrafico>.
- Rhapsody. 2014.** Diseño de casos de prueba - Procesos Productivos. [En línea] AngelFire, 2014. [Citado el: 28 de abril de 2014.] <http://www.angelfire.com/empire2/ivansanes/bywbox.htm>.
- Rodilla, Fernández, Abreu. 2013.** Educación especial y TIC. [En línea] 2013. <http://dcadepua.wikispaces.com/EDUCACI%C3%93N+ESPECIAL+Y+TIC>.
- Rodríguez, Juan. 2012.** Accesibilidad en la Web. [En línea] 2012. <http://juandarodriguez.es/tutoriales/inyeccion-de-dependencias-en-symfony2/>.
- Sánchez, Guerra, Esther. 2008.** Patrón de comportamiento Strategy. [En línea] 2008. <http://arantxa.ii.uam.es/~eguerra/docencia/0809/07%20Strategy.pdf>.
- SentidoWeb. 2007.** [En línea] 08 de febrero de 2007. <http://sentidoweb.com/2007/02/08/freetts-sintetizador-de-voz.php>.
- Softeng. 2014.** Metodología SCRUM para desarrollo de software a medida. [En línea] 2014. <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>.
- Solís, Calero, Manuel. 2003.** Una explicación de la programación extrema (XP). [En línea] 2003. <http://www.apolosoftware.com/>.

**Tedeschi, Nicolás. 2014.** ¿Qué es un Patrón de Diseño? [En línea] 2014. <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

**Universidad de Endinburgh. 2014.** The Festival Speech Synthesis System. [En línea] 2014. <http://www.cstr.ed.ac.uk/>.

**Vignoni, David. 2007.** VozMe. [En línea] 01 de 2007. <http://www.icon-king.com>.

**Visconti, Marcello y Astudillo, Hernán. 2014.** Fundamentos de Ingeniería de Software. [En línea] 2014.

**W3C. 2014.** Accessible Rich Internet Applications (WAI-ARIA) 1.0. [En línea] 20 de marzo de 2014. <http://www.w3.org/TR/wai-aria/>.

—. 2010. W3C. [En línea] 16 de septiembre de 2010. <http://www.w3.org/TR/wai-aria-primer/>.

**WebAcademia. 2013.** La síntesis de voz, Historia, Tecnologías Sintetizador, Retos, Hardware dedicado, Síntesis de voz microordenador, Mattel, SAM, Discurso lenguajes de marcado de síntesis, Aplicaciones. [En línea] 2013. [http://centrodeartigos.com/articulos-informativos/article\\_60373.html](http://centrodeartigos.com/articulos-informativos/article_60373.html).

**WebIntenta. 2014.** blog acerca de diseño, desarrollo web, recursos, fotografía, ilustración,.. [En línea] 2014. <http://webintenta.com/phpStorm-ide-para-php.html>.

**Wordpress. 2013.** Wordpress. [En línea] 2013. <http://es.wordpress.org/>.

## Anexos

### Anexo # 1 Instalación del componente.

A continuación se describen los pasos a realizar para instalar y configurar el componente. Es válido recordar que este componente proporciona una integración con una herramienta externa que se encarga de sintetizar el texto a audio. En este caso se muestra para el sintetizador de texto Festival, el cual es libre y de código abierto. En su sitio web se encuentran disponibles las versiones para los sistemas operativos GNU/Linux y Microsoft Windows. El presente anexo está centrado en Ubuntu ya que es el sistema operativo que se fomenta en la migración dentro de la Universidad de Ciencias Informáticas, además de ser Linux la plataforma recomendada para desarrollos web con Symfony 2.

#### Paso 1: Instalar FESTIVAL en Ubuntu:

```
sudo apt-get install festival
```

##### 1. Instalar las voces por defecto:

- a. Descomprimir el compactado de las voces.
- b. Abrir una consola e ir a esa carpeta y ejecutar: `sudo dpkg -i *.deb`

##### 2. Modificar el archivo de las voces de festival para indicar que use las nuevas voces:

- a. `sudo gedit /etc/festival.scm`
- b. Agregar al final (si no las tiene) las siguientes líneas:

```
(set! voice_default 'voice_JuntaDeAndalucia_es_pa_diphone)
```

```
(set! voice_default 'voice_JuntaDeAndalucia_es_sf_diphone)
```

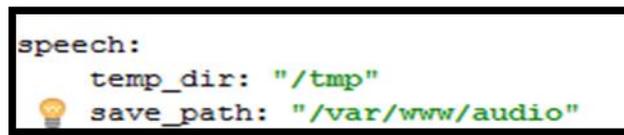
##### 3. Probar que el festival esté funcionando, se abre una consola y se ejecuta:

```
echo "texto de prueba" | iconv -f utf-8 -t iso-8859-1 | festival --tts
```

#### Paso 2: Instalar la aplicación:

El componente es un bundle de Symfony 2, por lo cual los pasos para instalarlo son prácticamente los mismos que para cualquier otro. Se asume que los parámetros de conexión a la base de datos ya han sido establecidos en el fichero `app/parameters.yml`:

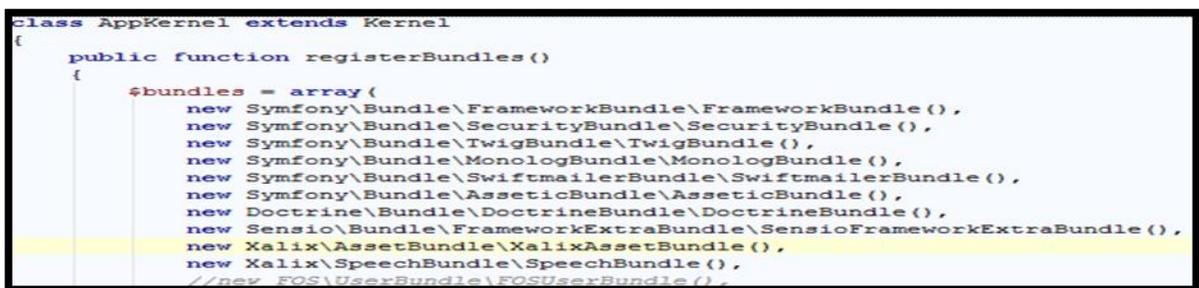
1. Copiar la carpeta `SpeechBundle` dentro de `src/Xalix`.
2. Editar el archivo `app/config.yml` y agregar los parámetros como se muestra a continuación:



```
speech:
  temp_dir: "/tmp"
  save_path: "/var/www/audio"
```

Figura 10. Agregar parámetros

- a. `temp_dir`: directorio temporal utilizado por el componente para sintetizar el texto. Debe tener permisos de lectura/escritura.
  - b. `save_path`: directorio donde se guardan los archivos de audio sintetizados por el componente. Este directorio debe existir y tener permisos de lectura/escritura por el sistema y ser accesible por el servidor web.
3. Registrar el bundle en `app/AppKernel.php`:
    - a. Para esto se añade la línea `new Xalix\SpeechBundle\SpeechBundle()`, dentro del arreglo de bundles, tal como se muestra en la figura 11:



```
class AppKernel extends Kernel
{
    public function registerBundles()
    {
        $bundles = array(
            new Symfony\Bundle\FrameworkBundle\FrameworkBundle(),
            new Symfony\Bundle\SecurityBundle\SecurityBundle(),
            new Symfony\Bundle\TwigBundle\TwigBundle(),
            new Symfony\Bundle\MonologBundle\MonologBundle(),
            new Symfony\Bundle\SwiftmailerBundle\SwiftmailerBundle(),
            new Symfony\Bundle\AsseticBundle\AsseticBundle(),
            new Doctrine\Bundle\DoctrineBundle\DoctrineBundle(),
            new Sensio\Bundle\FrameworkExtraBundle\SensioFrameworkExtraBundle(),
            new Xalix\AssetBundle\XalixAssetBundle(),
            new Xalix\SpeechBundle\SpeechBundle(),
            //new FOS\UserBundle\FOSUserBundle(),
        );
    }
}
```

Figura 11. Registrar bundle

4. Crear la BD del componente, abrir consola en la carpeta del componente y ejecutar:

- a. `php app/console doctrine:schema:update --force`

De esta manera se agregarán a la base de datos configurada las tablas utilizadas por el componente.

#### 5. Publicar los assets:

- a. `php app/console assets:install`

#### 6. Para utilizar el componente, basta con agregar en cualquier vista la siguiente función:

```
{{ speech_toolbar() }}
```

Esta función de twig se encarga de visualizar el componente al final de la página. También se deben cargar los javascripts y css correspondientes:

```
<link rel="stylesheet" href="{{ asset('bundles/speech/css/styles.css') }}" />
```

Figura 12. Cargar estilos css

```
<script type="text/javascript" src="{{ asset('bundles/speech/js/jquery.selection.js') }}"></script>  
<script type="text/javascript" src="{{ asset('bundles/speech/js/functions.js') }}"></script>
```

Figura 13. Cargar ficheros javascript

#### 7. El resultado final se muestra a continuación:



Figura 14. Interfaz del componente

### Anexo # 2 Implementación e integración de nuevos sintetizadores de texto a voz al componente.

El componente permite la integración con otros sintetizadores de texto a voz, para ello se deben implementar algunas clases y registrarlo como servicio.

Básicamente este proceso cuenta con dos fases fundamentales:

- 1- Se implementan las clases requeridas para que el componente se comunice con el sintetizador de texto a voz, el cual es una herramienta externa (ejemplo: FestivalTTS, eSpeak, etc). Luego se registra como servicio para que esté disponible para todo el sistema.
- 2- Se registra en la configuración (a través de una interfaz web) el nuevo sintetizador. Se adicionan a la base de datos los datos de las voces disponibles y se activa el sintetizador para que sea usado por el sistema.

A continuación se describen ambos procedimientos:

### Implementación de las clases requeridas:

1. Crear una nueva clase dentro del directorio SpeechBundle/TTS\_DRIVERS que representará al sintetizador que se desea integrar, por ejemplo SampleTTS.php, por convención se recomienda agregarle el sufijo TTS a estas clases para diferenciarlas de las demás.
2. La nueva clase (SampleTTS) debe heredar de Classes/AbstractSynthesizer. Ver el ejemplo de la figura 15:

```
namespace Xalix\SpeechBundle\TTS_Drivers;
use Xalix\SpeechBundle\Classes\AbstractSynthesizer;

class SampleTTS extends AbstractSynthesizer
{
    public function normalize($text, $lang)
    {}

    public function synthesize($text, $voice, $options = null)
    {}

    public function getFileExtensions()
    {}
}
```

**Figura 15. Clase de ejemplo**

3. Se deben implementar los métodos:
  - a. **normalize(\$text, \$lang)**: para normalizar el texto usando una herramienta externa al sintetizador.
  - b. **synthesize(\$text, \$voice, \$options = null)**: en este método se implementa la síntesis de texto a voz usando el sintetizador que se desea agregar, generalmente se hace ejecutando un comando desde el php con los parámetros y el texto. Debe retornar el nombre del fichero de audio sintetizado con el formato: <<nombre\_fichero>>.<<extensión>> . Deben

utilizarse los directorios temp y save\_path definidos en app/config.yml. Estas propiedades están disponibles a partir de la clase padre AbstractSynthesizer.

- c. **getFileExtensions:** devuelve un arreglo con las extensiones que soporta el sintetizador.  
Ejemplo: return array("wav","mp3");

4. Registrar la nueva clase como servicio:

- a. Se edita el fichero services.yml del bundle:



Figura 16. Fichero services.yml

- b. Se agrega la clase como un nuevo servicio al final del fichero:

```
services:
    speech.twig.extension:
        class: Xalix\SpeechBundle\Twig\SpeechExtension
        arguments: [ @service_container ]
        tags:
            - { name: twig.extension }
    speech:
        class: Xalix\SpeechBundle\Classes\Speech
        arguments: [ @doctrine.orm.entity_manager, @service_contai
    tts.espeak:
        class: Xalix\SpeechBundle\TTS_Drivers\EspeakTTS
    tts.festival:
        class: Xalix\SpeechBundle\TTS_Drivers\FestivalTTS
    tts.sample:
        class: Xalix\SpeechBundle\TTS_Drivers\SampleTTS
```

Figura 17. Agregar clase como un servicio

De esta forma queda integrado en el componente el nuevo sintetizador. Se debe destacar que el texto consecutivo al prefijo tts es el nombre con el cual se debe registrar en la base de datos, en este caso por ejemplo sería Sample.

Para mayor información sobre cómo implementar esta clase, se recomienda revisar el código de la clase FestivalTTS o EspeakTTS y utilizarlo como guía.

### Registrando en la base de datos el nuevo sintetizador implementado:

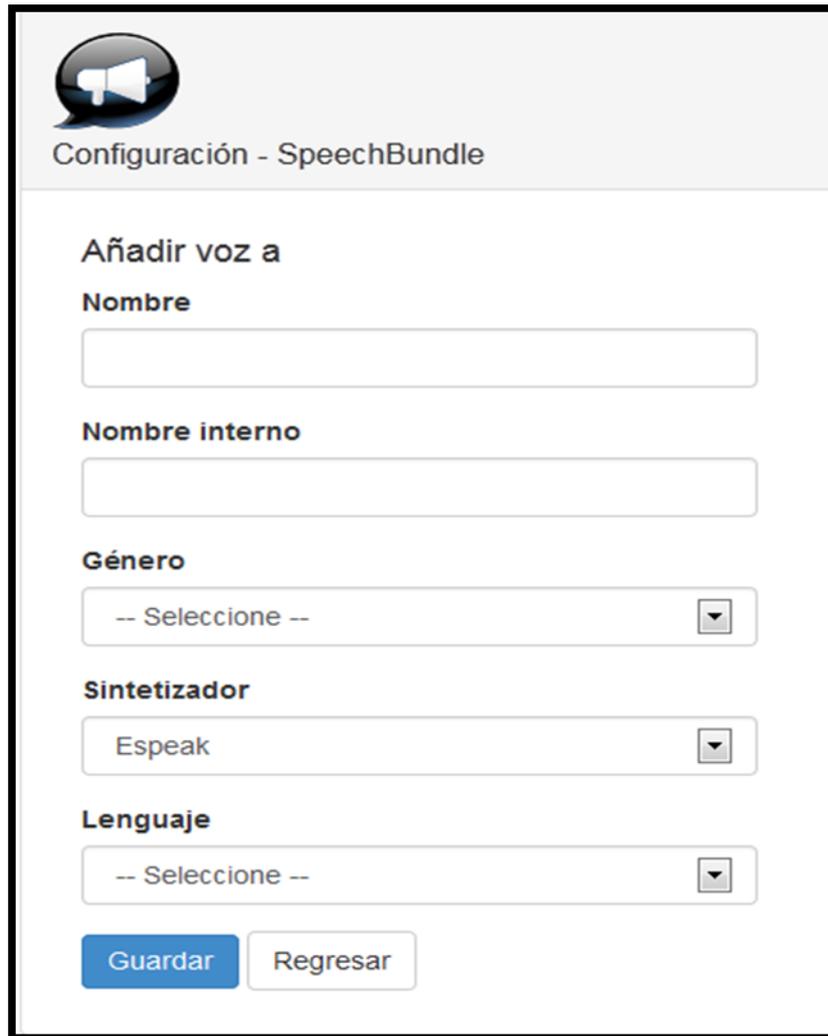
Para añadir el nuevo sintetizador de texto se accede a configuración, se presiona el botón “Añadir sintetizador” y se rellenan los campos que se muestran en la figura 18. El nombre del sintetizador tiene que coincidir con el nombre del servicio declarado en el archivo services.yml del bundle, eliminando el prefijo tts; pues el sistema verifica su existencia y muestra un mensaje de error en caso de no existir. Ejemplo para agregar el sintetizador de texto Sample, definido como servicio tts.sample, se agregaría a la base de datos solamente el texto sample. Además solo puede estar activo un sintetizador de texto a la vez.

The image shows a web interface titled "Configuración - SpeechBundle". At the top left is a speaker icon. Below the title, the section is labeled "Añadir nuevo sintetizador". There is a text input field labeled "Nombre". Below the input field is a checkbox labeled "Activo". At the bottom of the form are two buttons: "Añadir" (highlighted in blue) and "Regresar".

Figura 18. Añadir nuevo sintetizador

### Registrando en la base de datos las voces disponibles para el sintetizador:

Para añadir las voces del nuevo sintetizador de texto se accede a configuración, se presiona el botón “Añadir voz” y se rellenan los campos que se muestran en la figura 19. De los campos a llenar, el de mayor importancia es el campo nombre interno, ya que es el identificador con el cual el sintetizador de texto selecciona la voz a utilizar. Por ejemplo en el caso del sintetizador Espeak la voz español-latinoamericano tiene nombre interno es-la.



The image shows a web-based configuration interface titled "Configuración - SpeechBundle". At the top left is a speaker icon. Below the title, there are several form fields for adding a voice:

- Añadir voz a**: A section header.
- Nombre**: A text input field.
- Nombre interno**: A text input field.
- Género**: A dropdown menu with the text "-- Seleccione --".
- Sintetizador**: A dropdown menu with the text "Espeak".
- Lenguaje**: A dropdown menu with the text "-- Seleccione --".

At the bottom of the form are two buttons: "Guardar" (Save) and "Regresar" (Back).

Figura 19. Adicionar voces a un sintetizador de texto

### Anexo # 3 Etiquetas áreas para el funcionamiento del lector de pantalla

Las personas con ciertos tipos de deficiencias usan tecnologías asistidas (AT por sus en inglés) para interactuar con estos contenidos. Las AT pueden transformarse en la presentación de contenidos dentro de muchos formatos sustituibles para el usuario, incluso pueden interactuar con diferentes métodos. Por ejemplo, en un párrafo donde es semánticamente identificado que se utilizará tecnologías asistidas para interactuar con él, es únicamente separable para el resto de los contenidos conociendo su límite exacto.

La incorporación de WAI-ARIA<sup>14</sup>, definen propiedades comunes de APIs para la accesibilidad a plataformas que utilizan tecnologías asistidas usadas para proveer al usuario con efectivas interacciones y presentaciones. Las taxonomías de rol incluyen la interacción con *widgets* y elementos denotando estructuras de documentos. Estas taxonomías describen herencias y detalles de los atributos que soportan los roles.

Los roles son tipos de elementos y no cambiarán con el tiempo o acciones de usuarios. La información de los roles es utilizada por AT, con verdadera interacción de agentes de usuarios, que proveen un procesamiento normal de tipos de elementos especificados.

## **Roles**

En esta sección se define la taxonomía de los roles de WAI-ARIA y se describen las características y propiedades de todos los roles.

**article:** es utilizado en secciones de páginas que conforman partes independientes de un documento, una página o un sitio. Ejemplo para dar seguimiento a los envíos de comentarios en un foro, una revista, un periódico una entrada a la web.

**columnheader:** puede ser utilizado en el encabezado de una tabla, una cuadrícula o para trazar el pie de un mapa para mostrar una relación similar de un dato. Establece una relación entre una célula y sus correspondientes columnas. En HTML es una estructura equivalente al elemento `th` con una columna abarcada, por lo que se debe asegurar que los elementos con el rol *columnheader* son contenidos en, o tienen la propiedad de un elemento con el rol *row*.

**dialog:** es una ventana de aplicación que es diseñada para interrumpir el proceso actual de una aplicación en orden para disponer de información entrada por el usuario o requiere una respuesta. Se ve relacionado con *alertdialog*. Se debe proveer una etiqueta de diálogo.

---

<sup>14</sup> WAI -ARIA es una especificación técnica que proporciona un marco para mejorar la accesibilidad e interoperabilidad de los contenidos y aplicaciones web. Está diseñado para proporcionar información semántica sobre los objetos cuando los lenguajes principales carecen de una semántica nativa para el objeto.

**directory:** en una lista de referencias para miembros de un grupo, que están en una tabla de contenidos estática, tanto vinculada como desvinculada, se debe utilizar este rol. Es requerido en tablas de contenidos construidas con listas, incluyendo listas anidadas. Las tablas dinámicas de contenidos, sin embargo, fuerzan a usar el rol **tree** en su lugar.

**document:** es utilizado cuando el usuario navega a un elemento. Este rol informa la necesidad de aumentar el soporte del teclado del navegador con el fin de permitir visitar y leer cualquier contenido dentro de la región definida por *document*.

Para un correcto funcionamiento de un lector de pantalla se deben proporcionar un título o una etiqueta para los documentos. Se debería usar un texto adecuado para el uso de la etiqueta, como una vista previa de navegación o entrada de la tabla - de - contenido para la sección de la página. Las etiquetas deben ser utilizadas a través de los siguientes métodos:

- ✓ Si el documento incluye todo el contenido de la página web, utilizar la función de lengua de acogida para el título o en la etiqueta, como el elemento *title* en HTML y SVG. Esto permite etiquetar todo el documento.
- ✓ Proporcionar una etiqueta visible que haga referencia el documento utilizando *aria-labelledby*.

**group:** es utilizado en un cambio de objetos de interfaces de usuarios que no serán destinados a incluirse en el resumen de páginas o en tablas de contenidos por tecnologías asistidas. Contrasta con el rol *región* que no es más que un agrupamiento de estos objetos de interfaces de usuarios.

Se debe usar *group* para coleccionar una forma lógica de artículos en un *widget*, tales como sus nodos hijos en un árbol formando en una jerarquía, una colección de nodos hermanos, o una colección de artículos que tienen algún directorio contenido. Por lo que, si una sección es suficientemente significativa como para justificar su inclusión en el cuadro de contenidos de la página web, se debe asignar la sección de un rol *region* o un rol *standard landmark*.

**heading:** a menudo los elementos *heading* serán referenciados con el atributo *aria-labelledby* de la sección para que sirvan como títulos. Si los títulos son organizados dentro de un esquema lógico, el atributo *aria-level* puede ser usado para indicar el nivel de anidación.

**img:** para que los elementos con rol *img* sean perceptibles, se debe propiciar un texto alternativo o una etiqueta determinada por el *accessible name calculation*.

**list:** se utiliza en un grupo de elementos de listas no iterativas. Sus nodos hijos requieren el rol *listitem* o elementos cuyo rol es *group* que a su vez tengan nodos hijos que contengan el rol *listitem*.

**listitem:** es aplicado a un solo elemento de una lista o de un directorio. Se debe asegurar elementos con este rol cuando son contenidos en, o tienen como propiedad un elemento con el rol *list* o *group*.

**match:** es aplicado a contenidos que representan una expresión matemática. Los contenidos con este rol son entendidos para ser construidos dentro de un formato entendible tales como *MathML [MATHML]*, o con otros tipos de representaciones textuales como TeX o LaTeX, que puedan ser convertidos para un formato accesible por tecnologías asistidas.

**note:** es utilizado en secciones cuyo contenido sea entre paréntesis o auxiliar para la página principal del recurso.

**presentation:** es requerido para un elemento cuyo rol de semántica nativa implícita no se asignará a ninguna API. Principalmente se usa cuando un elemento cambia la vista de una página pero no tiene toda la relevancia funcional, interactiva o estructural por el tipo de elemento, o tal vez para proveer de una reserva en otros navegadores que no soportan WAI-ARIA.

**region:** es utilizado en partes perceptibles de una página web o un documento, que son lo suficientemente importantes como para ser incluidos en un resumen de una página o en una tabla de contenidos. Se debe asegurar que una región tenga un título referenciado por *aria-labelledby*. Este título es provisto por una instancia de un lenguaje acogedor de elementos de títulos o una instancia de un elemento con rol *heading* que contiene el texto del título.

**row:** las filas contienen elementos *gridcell*, y así permiten organizar las cuadrículas. En un árbol de cuadrículas, se pueden marcar filas expandibles, usando el atributo *aria-expanded* para indicar el estado presente. En cuadrículas ordenadas, no funciona.

Para este caso es necesario un elemento con el rol *row* que son contenido en, o provistos por un elemento con los roles *grid*, *rowgroup*, *treegrid*.

***rowgroup***: establece una relación entre elementos con propiedades de filas. Esto es una equivalencia estructural para elementos *thead*, *tfoot* y *tbody* en elementos de tabla HTML.

***rowheader***: puede ser usado en la cabecera de una fila en una tabla o una cuadrícula. Este rol establece una relación entre la cabecera y todas las celdas correspondientes a la fila. Lo cual es equivalente a la configuración `scope="row"` o al elemento `th` en HTML.

***separator***: es utilizado como separador para distinguir secciones de contenidos o grupos de elementos de menú. Es un separador visual entre secciones de contenidos. Por ejemplo, los separadores son encontrados entre grupos de elementos de menú y un menú, o donde el separador se mueva entre 2 regiones un panel dividido.

***toolbar***: representa una colección de uso común en botones o formularios de controles representados en compactados visual. Es utilizado en *menubar*, diseñado para reducir el esfuerzo del usuario para utilizar estas funciones. Donde se utilicen muchos roles *toolbar*, se debe suministrar una propiedad *aria-label* para cada uno.

***input***: es un tipo de *widget* genérico que permite la entrada de usuarios.

***select***: es abstracto, y usado por ontología. Consiste en un formulario *widget* que permite al usuario hacer selecciones a partir de un conjunto de opciones. No se debe utilizar en el rol *contenido*.

***widget***: es un componente interactivo de una interfaz gráfica de usuarios (GUI). Consiste en objetos de interfaces discretas con las que puede interactuar el usuario. Cuando el usuario navega en elementos que tienen cualquier rol de *widget* de subclases no abstractas asignadas, las tecnologías asistidas deben cambiar a un modo de navegación de aplicaciones y pasar eventos de teclado a través de la aplicación web. No se debe utilizar en el rol *contenido*.

**link:** una referencia interactiva para un recurso interno o externo que, cuando se active permita la navegación por el recurso. Si al pulsar el enlace se activa una acción, pero no cambia el foco del navegador o la ubicación de esta página, se recomienda considerar el uso del rol *botón* en lugar del rol *link*.

**Anexo # 4 Descripciones de los casos de uso**

**Tabla 9. CU Permitir la normalización texto**

CU Permitir la normalización texto	
<b>Objetivo</b>	Permitir que se normalice un texto, ya sea desde un sintetizador o desde una herramienta externa.
<b>Actores</b>	<ul style="list-style-type: none"> <li>✓ Usuario del sistema (Usuario)</li> <li>✓ Administrador del sistema (Administrador)</li> </ul>
<b>Resumen</b>	El caso de uso inicia cuando un texto es enviado a una herramienta encargada de normalizar texto. Una vez realizada la acción, se finaliza el caso de uso.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Alta
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>✓ Se ha accedido a un sitio web que incluye el componente para leer de pantallas</li> <li>✓ El componente ha capturado un texto</li> </ul>

<b>Postcondiciones</b>		Se normalizó un texto.
<b>Flujo de eventos</b>		
<b>Flujo básico “Permitir normalización de texto a través de un sintetizador”</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona reproducir texto o descargar archivo de audio del texto seleccionado.	
2		Envía el texto a un sintetizador de voz que internamente lo normaliza.
<b>Flujos alternos</b>		
2. a El sistema no cuenta con un sintetizador de voz que normaliza el texto internamente.		
	<b>Actor</b>	<b>Sistema</b>
1		No normaliza el texto
<b>Sección 1 “Permite normalizar texto a través de una herramienta externa”</b>		
<b>Flujo básico “Normalizar texto a través de una herramienta externa”</b>		

	<b>Actor</b>	<b>Sistema</b>
1	Selecciona reproducir texto o descargar archivo de audio del texto seleccionado.	
2		Envía el texto a una herramienta encargada de normalizar texto.
<b>Flujos alternos</b>		
2. a El sistema no cuenta con una herramienta para normalizar texto.		
	<b>Actor</b>	<b>Sistema</b>
1		No normaliza texto
<b>Relaciones</b>	<b>CU Incluidos</b>	<ul style="list-style-type: none"> <li>✓ Selecciona reproducir texto en el <b>CU Reproducir texto</b></li> <li>✓ Selecciona descargar texto en el <b>CU Descargar archivo de audio</b></li> </ul>

**Tabla 10. CU Permitir sintetizar a voz el texto mediante el uso de un sintetizador de voz**

**CU Permitir sintetizar a voz el texto mediante el uso de un sintetizador de voz**

<b>Objetivo</b>	Convertir el texto en un archivo de audio, de acuerdo con el idioma definido para el sitio.	
<b>Actores</b>	<ul style="list-style-type: none"> <li>✓ Usuario del sistema (Usuario)</li> <li>✓ Administrador del sistema (Administrador)</li> </ul>	
<b>Resumen</b>	El caso de uso inicia cuando un texto es enviado a un sintetizador de voz, encargado de convertir el texto en audio. Una vez sintetizado el texto finaliza el caso de uso.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>✓ Se ha accedido a un sitio web que incluye el componente para leer de pantallas</li> <li>✓ El componente ha capturado un texto</li> </ul>	
<b>Postcondiciones</b>	Se convirtió el texto en un archivo de audio, en concordancia con el idioma definido dentro del sitio.	
<b>Flujo de eventos: Permitir sintetizar a voz el texto mediante el uso de un sintetizador de voz</b>		
<b>Flujo básico "Sintetizar a voz el texto"</b>		
	<b>Actor</b>	<b>Sistema</b>

1	Selecciona reproducir texto o descargar archivo de audio del texto seleccionado.	
2		<ul style="list-style-type: none"> <li>✓ Captura el idioma definido para el sitio</li> <li>✓ Envía a un sintetizador de voz el texto y el idioma en que debe sintetizar el texto</li> </ul>
<b>Flujos alternos</b>		
2.a El sitio web no tiene definido el idioma		
	<b>Actor</b>	<b>Sistema</b>
1		El sintetizador utiliza el idioma español por defecto.
<b>Relaciones</b>	<b>CU Incluidos</b>	<b>CU Identificar el idioma del texto.</b>

**Tabla 11. CU Cambiar voz del sintetizador de voz**

<b>CU Cambiar voz del sintetizador de voz</b>	
<b>Objetivo</b>	Permitir que el usuario seleccione la voz que desea escuchar.
<b>Actores</b>	<ul style="list-style-type: none"> <li>✓ Usuario del sistema (Usuario)</li> <li>✓ Administrador del sistema (Administrador)</li> </ul>
<b>Resumen</b>	El caso de uso inicia cuando el actor del sistema selecciona la voz que desea escuchar. Una vez seleccionada la voz finaliza el

	caso de uso.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Baja	
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>✓ Se ha accedido a un sitio web que incluye el componente para leer textos de pantalla</li> <li>✓ El componente ha capturado un texto</li> </ul>	
<b>Postcondiciones</b>	Se seleccionó la voz para sintetizador de voz utilizado.	
<b>Flujo de eventos</b>		
<b>Flujo básico “Cambiar voz del sintetizador de voz”</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona la voz que desea escuchar	
2		Provee una interfaz que le permite seleccionar la voz que desea escuchar.
<b>Flujos alternos</b>		
1.a El actor no selecciona la voz que desea escuchar		
	<b>Actor</b>	<b>Sistema</b>
1		Define una voz por defecto.

**Tabla 12. CU Configurar opciones provistas por el sintetizador de voz**

**CU Configurar opciones provistas por el sintetizador de voz**

<b>Objetivo</b>	Permitir que el usuario configure las características específicas del sintetizador de voz disponibles para todo el sistema.	
<b>Actores</b>	Administrador del sistema (Administrador).	
<b>Resumen</b>	El caso de uso se inicia cuando el actor del sistema configura las características específicas de un sintetizador de voz, las cuales se aplican a todo el sistema. Una vez configurado el sintetizador de voz, finaliza el caso de uso.	
<b>Complejidad</b>	Normal	
<b>Prioridad</b>	Normal	
<b>Precondiciones</b>	El componente debe contar con un sintetizador de voz.	
<b>Postcondiciones</b>	Se configuró el sintetizador de voz.	
<b>Flujo de eventos</b>		
<b>Flujo básico: “Configurar opciones provistas por el sintetizador de voz”</b>		
	<b>Actor</b>	<b>Sistema</b>
1	El actor accede a la configuración del sintetizador de voz en el la ventana de administración del sistema.	
2		El sistema muestra una interfaz con las opciones disponibles para configurar el sintetizador de voz. Estas opciones pueden variar, dependiendo del sintetizador de voz utilizado.
3	El actor selecciona una de las opciones mostradas por el sistema.	

4		El sistema muestra los valores posibles para la opción seleccionada.
5	<ul style="list-style-type: none"> <li>✓ El actor modifica la opción seleccionada</li> <li>✓ El actor confirma los cambios</li> </ul>	
6		<ul style="list-style-type: none"> <li>✓ El sistema verifica la nueva configuración</li> <li>✓ El sistema aplica la modificación</li> </ul>
<b>Flujos alternos</b>		
5.a El actor introduce valores no válidos		
	<b>Actor</b>	<b>Sistema</b>
1		<ul style="list-style-type: none"> <li>✓ El sistema muestra un mensaje de error</li> <li>✓ El sistema no aplica la configuración</li> <li>✓ El sistema vuelve a la interfaz anterior</li> </ul>

**Tabla 13. CU Configurar opciones provistas por el sintetizador de voz**

<b>CU Capturar texto seleccionado por el usuario</b>	
<b>Objetivo</b>	Obtener el texto que un usuario seleccione del sitio web.
<b>Actor</b>	<ul style="list-style-type: none"> <li>✓ Usuario del sistema (Usuario)</li> <li>✓ Administrador del sistema (Administrador)</li> </ul>

<b>Resumen</b>	El caso de uso inicia cuando el actor selecciona un texto. El componente captura ese texto y así se finaliza el caso de uso.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	Se ha accedido a un sitio web que incluye el componente para leer de pantallas.	
<b>Postcondiciones</b>	Se obtuvo un texto seleccionado por el actor.	
<b>Flujo de eventos</b>		
<b>Flujo básico "Seleccionar texto "</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona un texto del sitio web.	
2		Captura el texto seleccionado.
<b>Flujos alternos</b>		
1. a El actor no selecciona ningún texto.		

	<b>Actor</b>	<b>Sistema</b>
1		Captura todo el texto del sitio web.

**Tabla 14. CU Descargar archivo de audio**

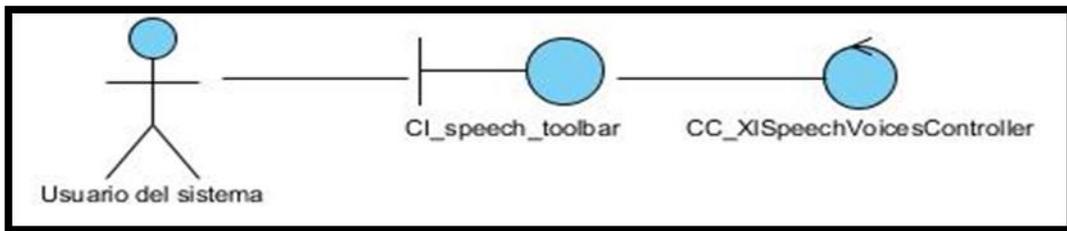
<b>CU Descargar archivo de audio</b>	
<b>Objetivo</b>	Descargar el archivo de audio de un texto, devuelto por un sintetizador de voz, dentro de un sitio web. Permitir el uso de atajos del teclado para la descarga del archivo de audio.
<b>Actores</b>	<ul style="list-style-type: none"> <li>✓ Usuario del sistema (Usuario)</li> <li>✓ Administrador del sistema (Administrador)</li> </ul>
<b>Resumen</b>	El caso de uso inicia cuando el actor del sistema selecciona un texto para descargarlo como audio. Una vez descargado el archivo, finaliza el caso de uso.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Normal
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>✓ El usuario accede a un sitio web que tenga instalado el componente</li> </ul>
<b>Postcondiciones</b>	Se descarga un archivo de audio de un texto X.

<b>Flujos de eventos</b>		
<b>Flujo básico “Descargar texto como archivo de audio”</b>		
	<b>Actor</b>	<b>Sistema</b>
1	El actor del sistema selecciona un texto.	
2		El sistema captura el texto seleccionado.
3	El actor del sistema accede a la descarga del archivo de audio del texto seleccionado.	
4		El sistema muestra una interfaz, que permite descargar el archivo de audio del texto de la siguiente manera: <ul style="list-style-type: none"> <li>✓ A través de un botón para la descarga.</li> <li>✓ A través del uso de atajos del teclado.</li> </ul>
5	El actor del sistema hace click en el botón para la descarga.	
6		El sistema envía al navegador el archivo para la descarga.
<b>Flujos alternos</b>		

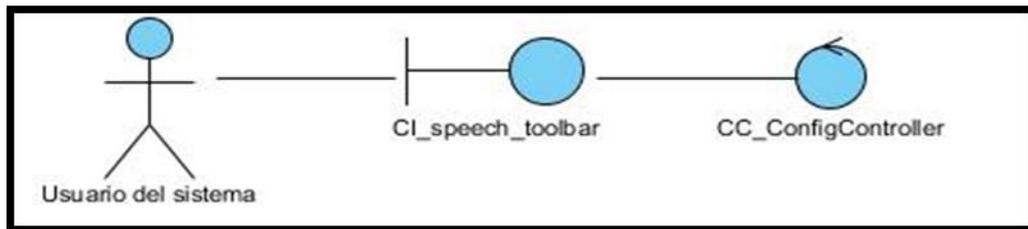
1. a El usuario no selecciona texto.		
	<b>Actor</b>	<b>Sistema</b>
1		El sistema captura todo el contenido del sitio web definido como relevante mediante las etiquetas <i>role</i> .
<b>Sección 1 “Permitir uso de atajo del teclado para la descarga”</b>		
<b>Flujo básico “Permitir uso de atajo del teclado para la descarga”</b>		
	<b>Actor</b>	<b>Sistema</b>
	El actor del sistema utiliza el atajo del teclado para la descarga.	
		<ul style="list-style-type: none"> <li>✓ El sistema comprueba el atajo de teclado utilizado</li> <li>✓ El sistema envía el fichero al navegador para efectuar la descarga.</li> </ul>
<b>Flujos alternos</b>		
7.a El atajo del teclado utilizado no es válido para la descarga		
	<b>Actor</b>	<b>Sistema</b>
1		El sistema no realiza ninguna acción.

<b>Relaciones</b>	<b>CU Incluidos</b>	<ul style="list-style-type: none"> <li>✓ CU Uso de atajos del teclado</li> <li>✓ CU Capturar texto</li> </ul>
-------------------	---------------------	---

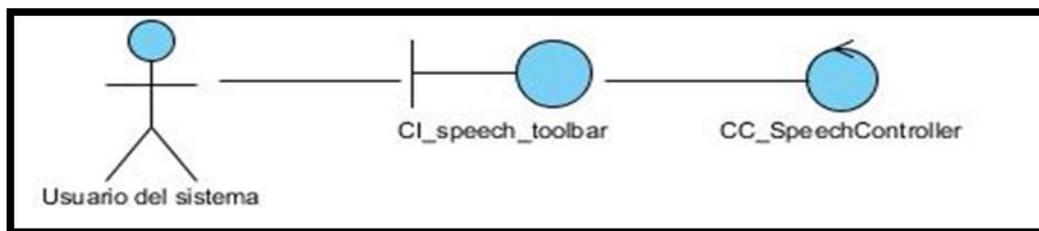
**Anexo # 5: Diagramas de clases del análisis**



**Figura 20. Diagrama de clases del análisis del CU Cambiar voz del sintetizador de voz**



**Figura 21. Diagrama de clases del análisis del CU Configurar opciones específicas del sintetizador de voz**



**Figura 22. Diagrama de clases del análisis del CU Descargar archivo de audio**

Anexo # 6 Diagramas de colaboración

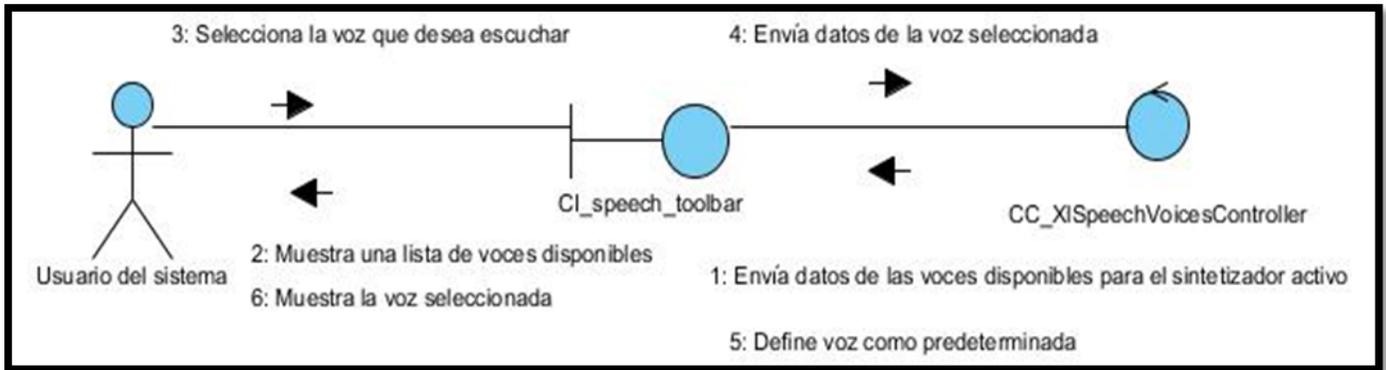


Figura 23. Diagrama de colaboración del CU Cambiar voz del sintetizador de voz

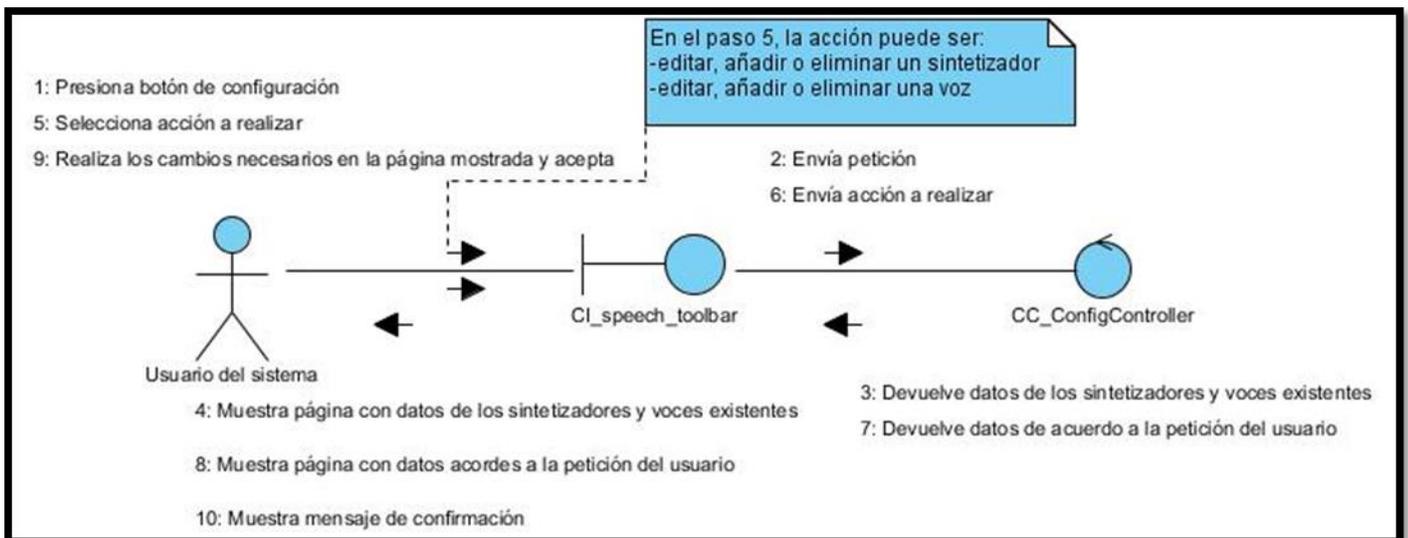
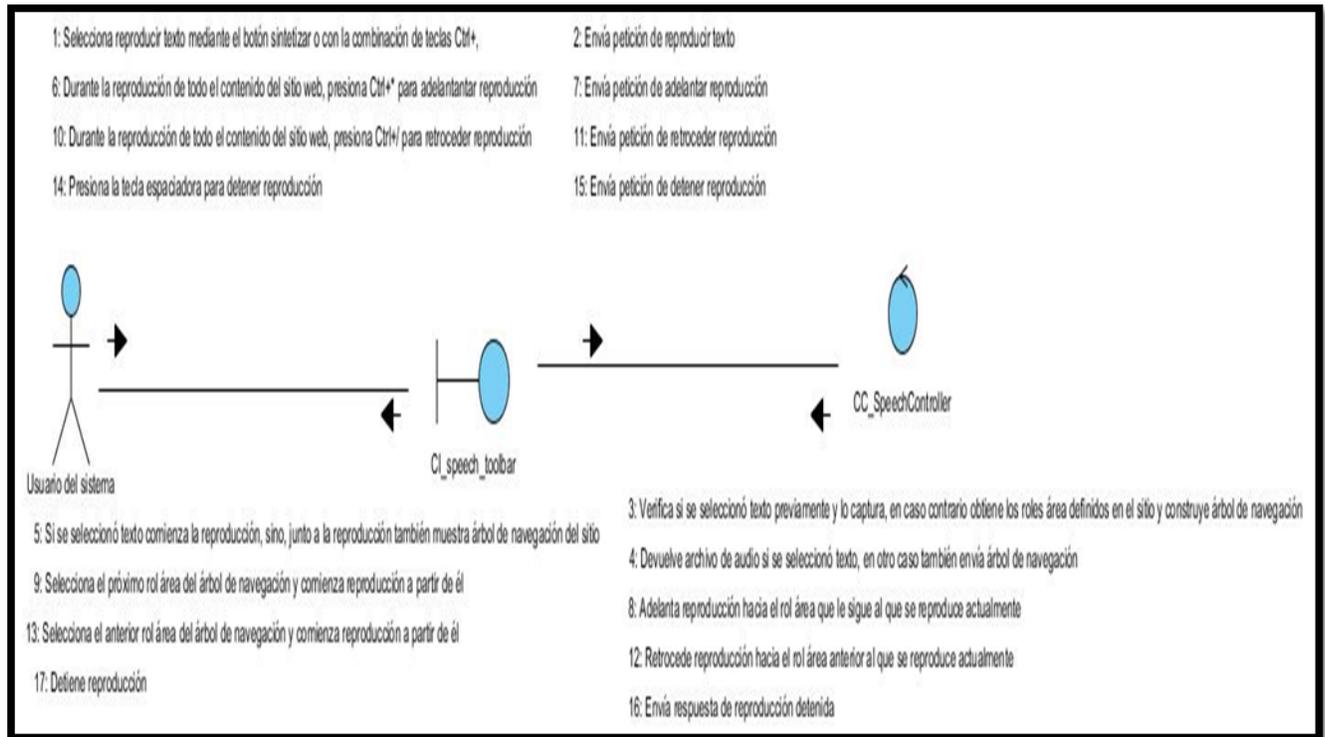
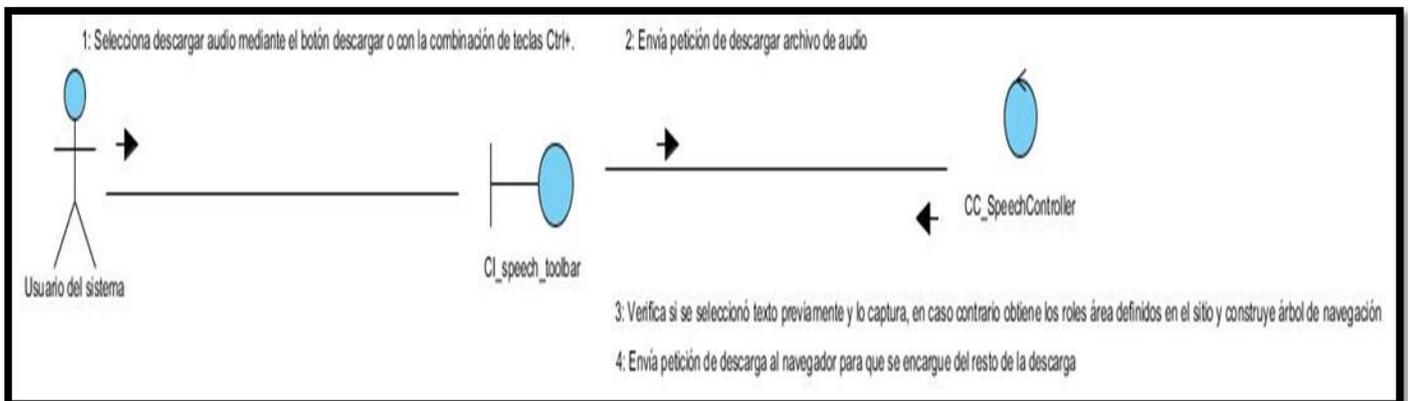


Figura 24. Diagrama de colaboración del CU Configurar opciones específicas del sintetizador de voz



**Figura 25. Diagrama de colaboración del CU Reproducir texto**



**Figura 26. Diagrama de colaboración del CU Descargar archivo de audio**

Anexo # 7 Diagramas de diseño de clases de los casos de uso

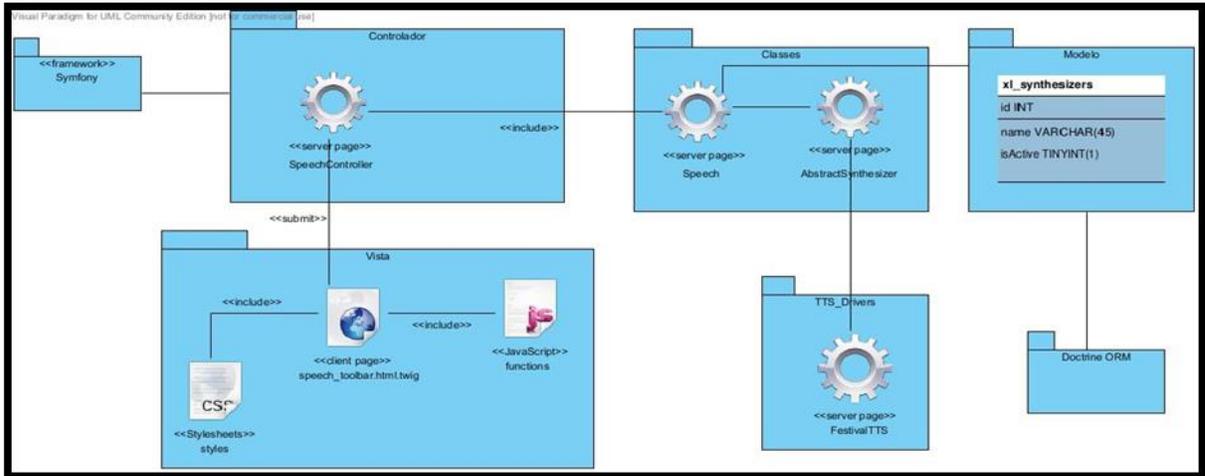


Figura 27. Diagrama de diseño de clases del CU Normalizar texto

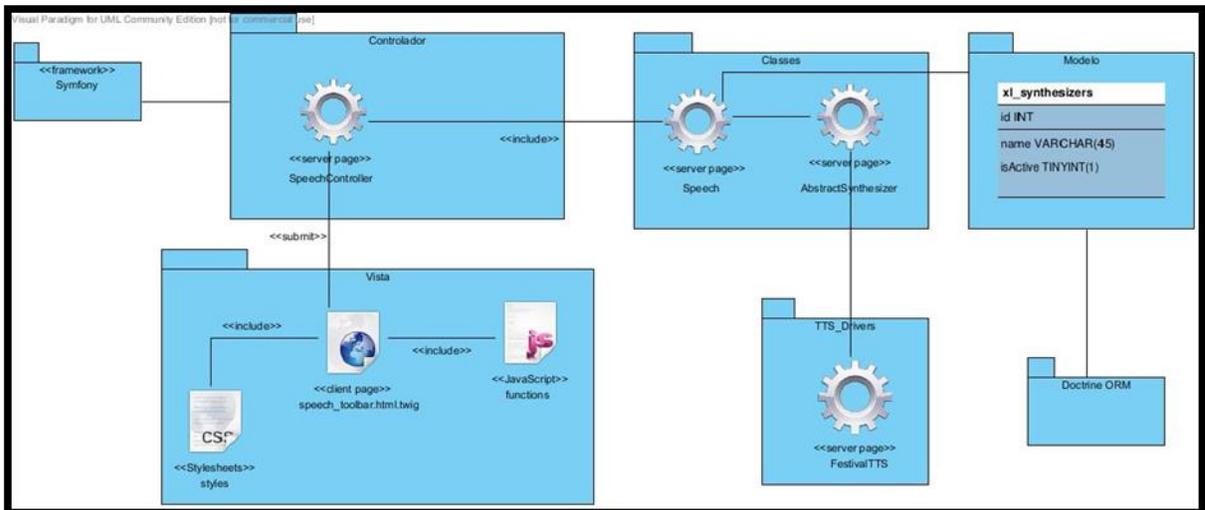


Figura 28. Diagrama de diseño de clases del CU Sintetizar a voz el texto

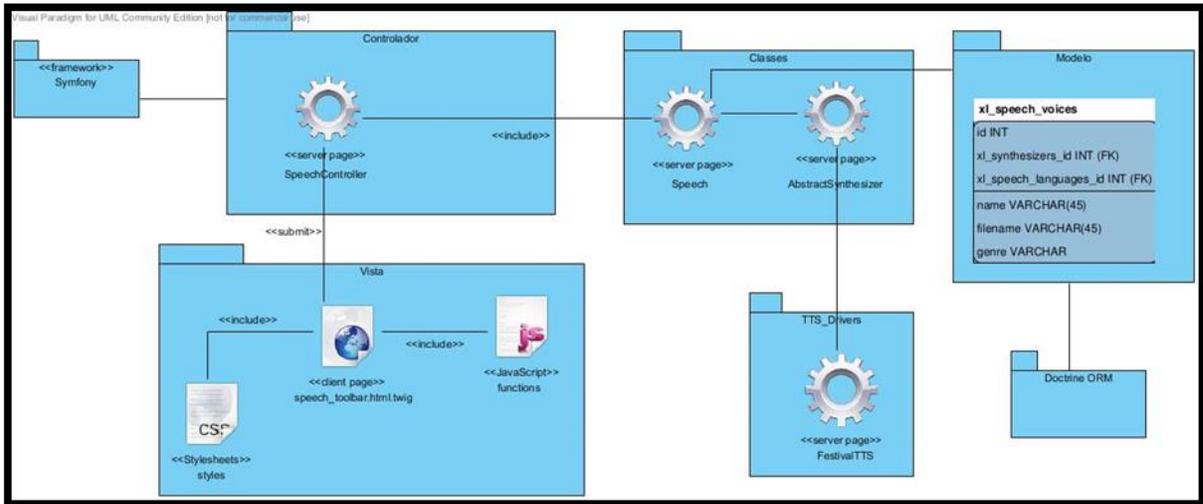


Figura 29. Diagrama de diseño de clases del CU Cambiar voz del sintetizador de voz

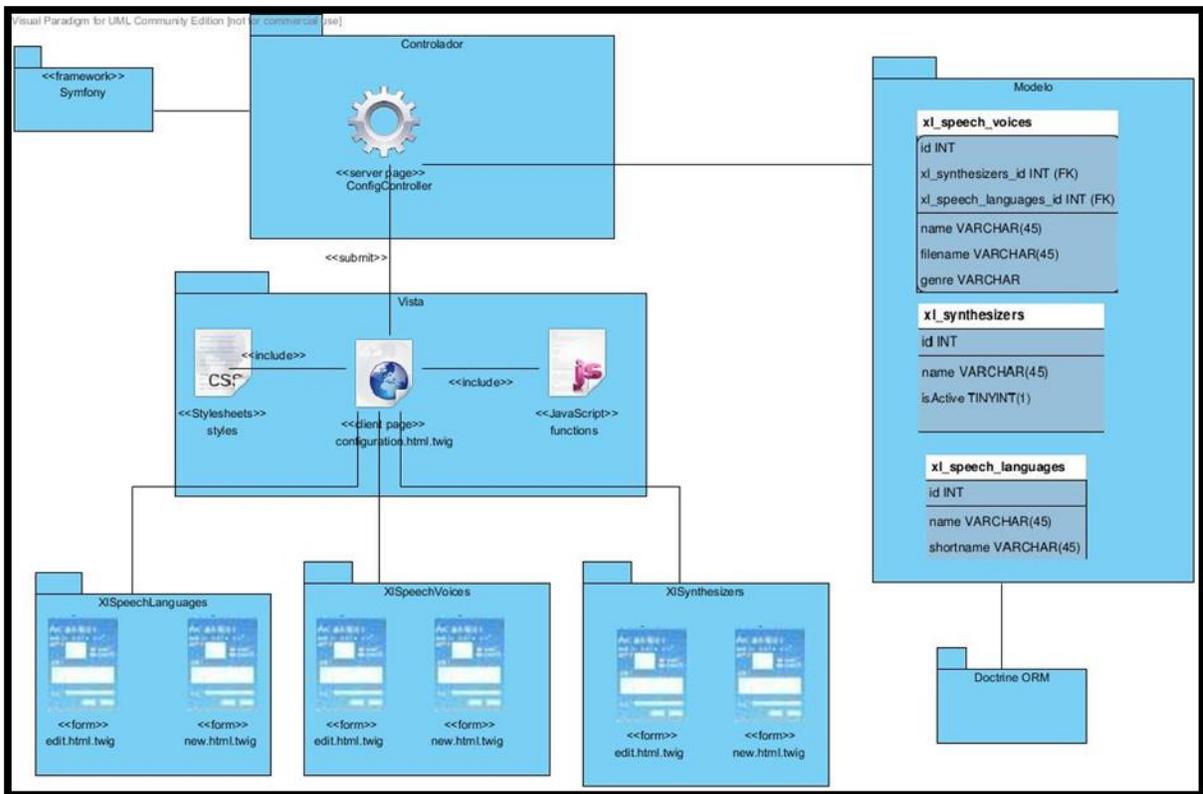


Figura 30. Diagrama de diseño de clases del CU Configurar opciones específicas del sintetizador de voz

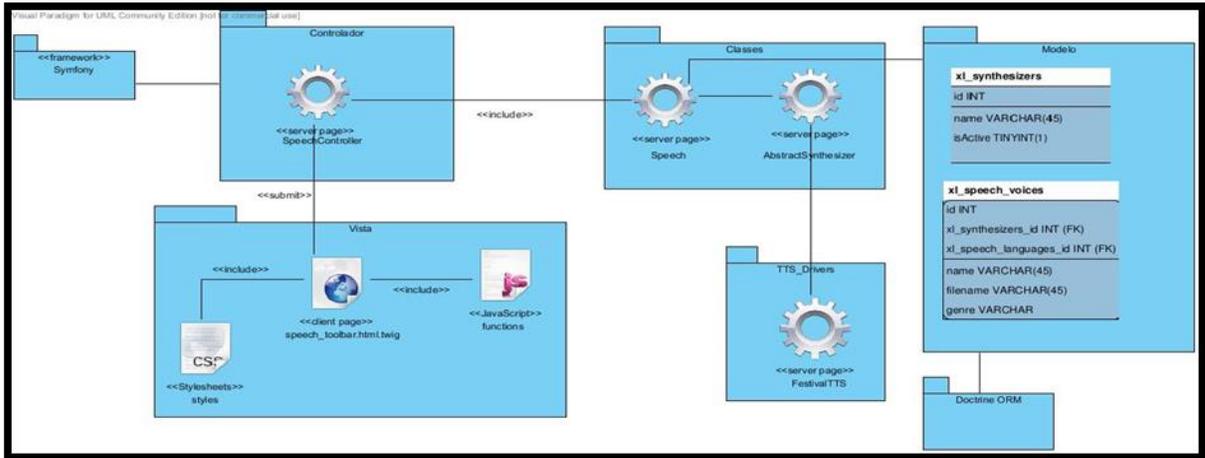


Figura 31. Diagrama de diseño de clases del CU Reproducir texto

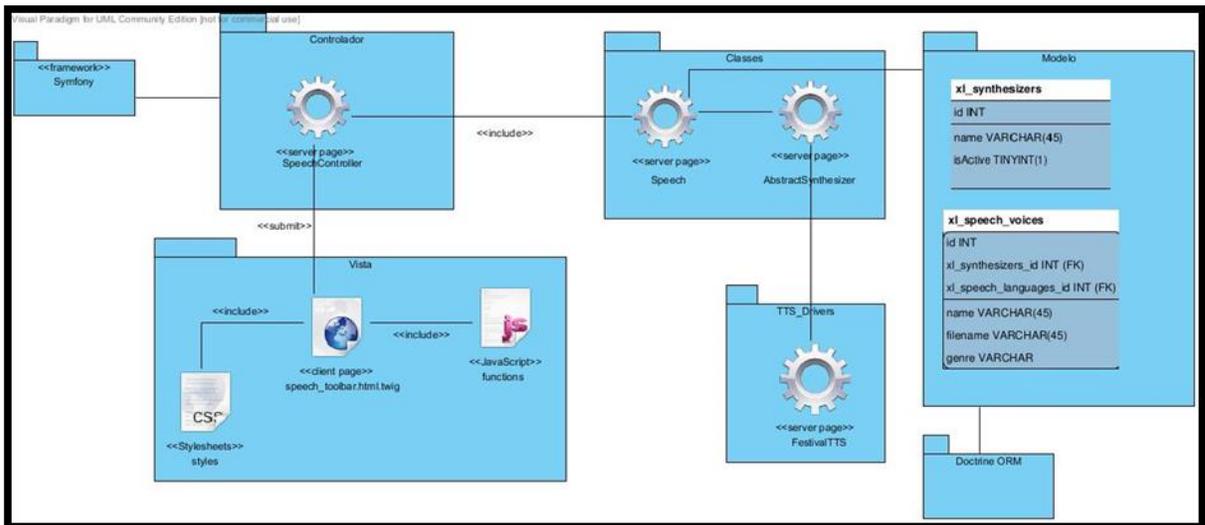


Figura 32. Diagrama de diseño de clases del CU Descargar archivo de audio

Anexo # 8 Diagramas de componentes

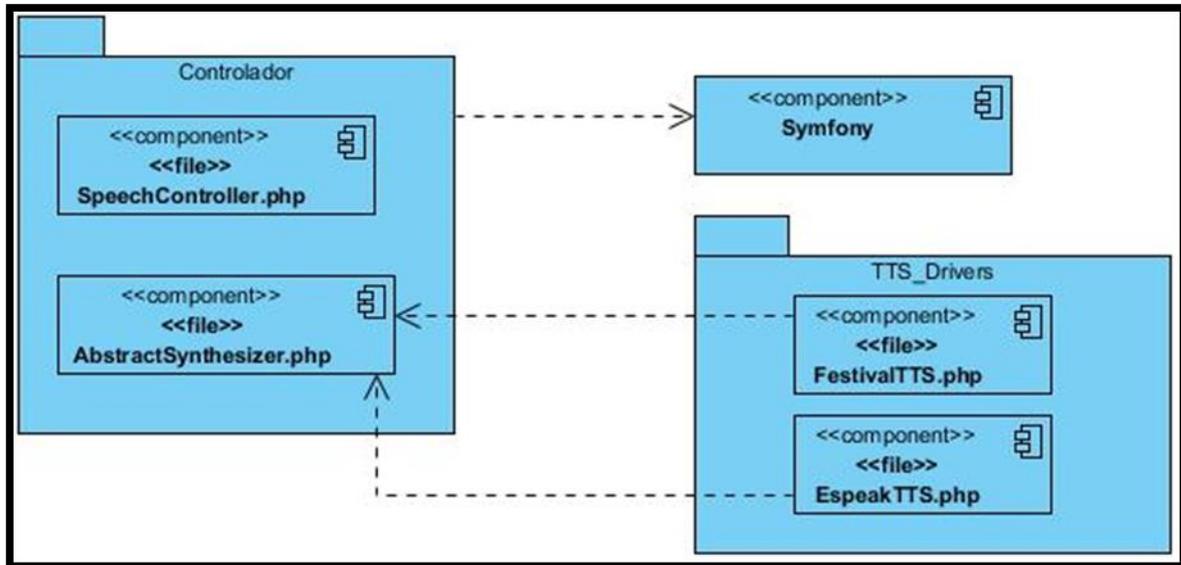


Figura 33. Diagrama de componentes de los CU Normalizar texto y Sintetizar a voz el texto

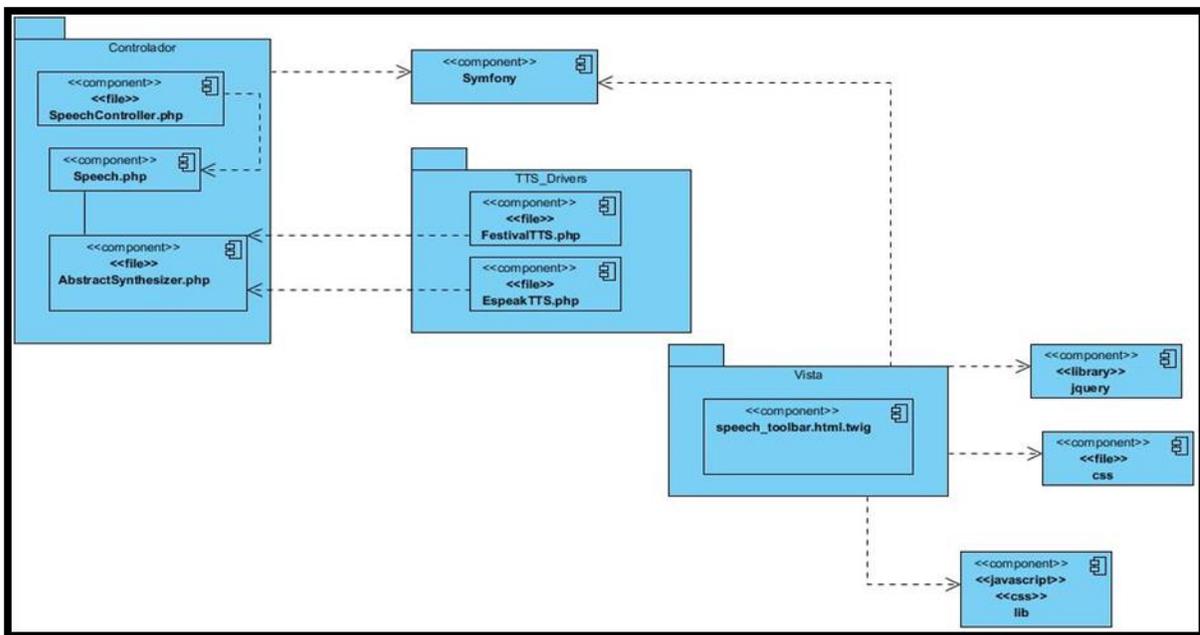


Figura 34. Diagrama de componentes del CU Cambiar voz del sintetizador de voz

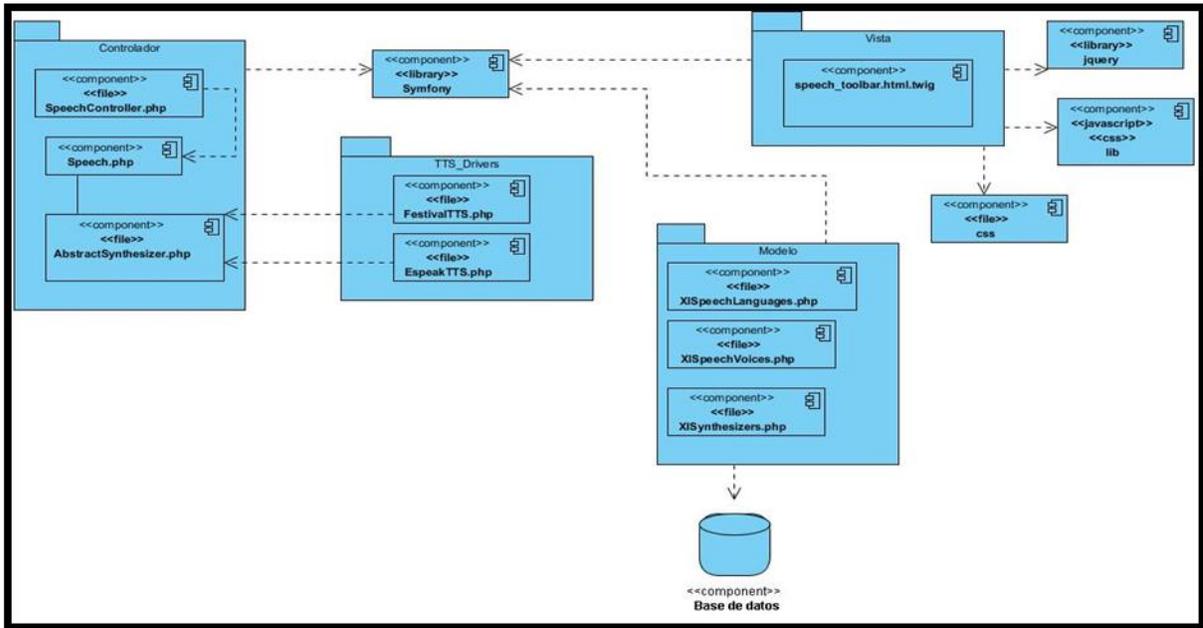


Figura 35. Diagrama de componentes del CU Configurar opciones específicas del sintetizador de VOZ

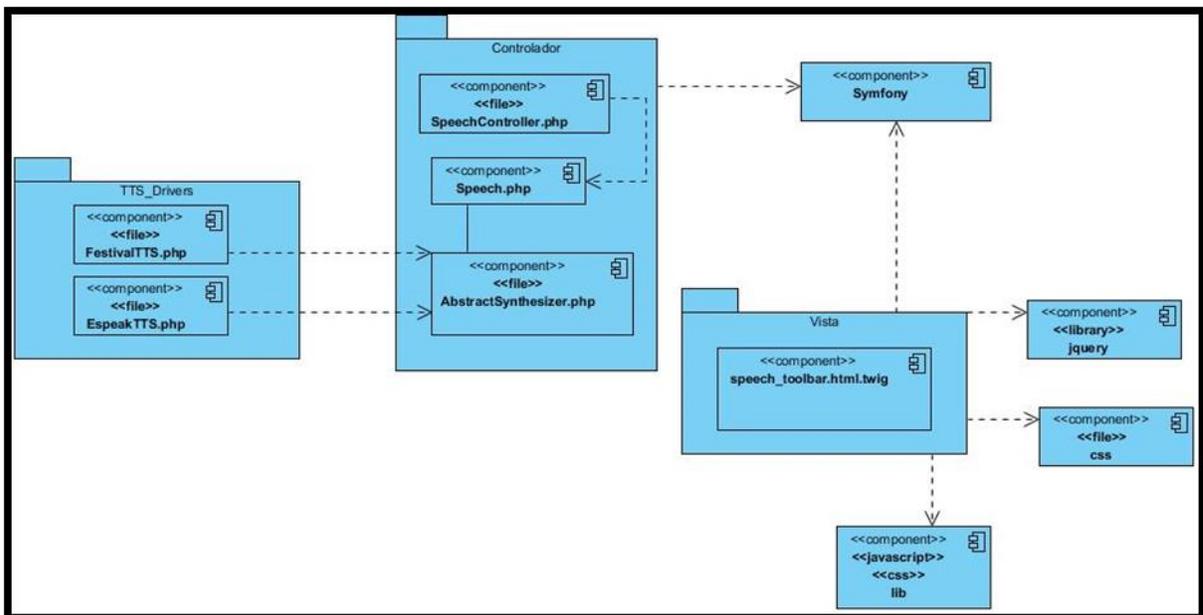


Figura 36. Diagrama componentes del CU Descargar archivo de audio

**Anexo # 9 Descripción de los casos de pruebas**

**Tabla 15. Descripción del CP Cambiar voz del sintetizador**

CP Cambiar voz del sintetizador				
Descripción general				
Se inicia cuando el actor del sistema selecciona una de las voces disponibles en el cuadro de selección. Una vez seleccionada la voz se finaliza.				
Condiciones de ejecución				
Se ha accedido a un sitio web que incluye el componente para leer de pantallas.				
El componente ha capturado un texto.				
SC1 Cambiar voz del sintetizador				
Escenario	Descripción	Nombre de la voz	Respuesta del sistema	Flujo central
EC 1.1	Selecciona una de las voces que tiene disponible el sintetizador.	V	Captura el identificador de la voz seleccionada. Toma esa voz como predeterminada para la reproducción del texto.	Para comprobar que la selección de la voz funciona correctamente se accede a una página web que contenga el componente y en la lista desplegable del componente se selecciona la voz a escuchar. Se mostrará en la lista esta voz como predeterminada. El texto solo se escuchará si la voz realmente pertenece al sintetizador de texto a voz activo en ese momento. En este caso específico se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.

**Tabla 16. Descripción del CP Configurar opciones provistas por el sintetizador**

CP Configurar opciones específicas del sintetizador
---

<b>Descripción general</b>								
Se inicia cuando el administrador del sistema configura las características específicas de un sintetizador de voz, las cuales se aplican a todo el sistema. Una vez configurado el sintetizador de voz se finaliza.								
<b>Condiciones de ejecución</b>								
El componente cuenta con al menos un sintetizador de voz.								
<b>SC1 Configuraciones generales</b>								
<b>Escenario</b>	<b>Descripción</b>	<b>Nombre del sintetizador</b>	<b>Sintetizador activo</b>	<b>Nombre de la voz</b>	<b>Nombre técnico de la voz</b>	<b>Permisos de administración</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC 1.1	Accede a una página web que tenga integrado el componente	N/A	N/A	N/A	N/A	V	Muestra el botón de configuraciones del sistema.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
		N/A	N/A	N/A	N/A	I	No muestra el botón de configuraciones del sistema	
EC 1.2	Presiona el botón de configuración del sistema	N/A	N/A	N/A	N/A	V	Muestra una ventana con los datos almacenados de los sintetizadores de textos que tiene el componente.	En este caso específico se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
<b>SC2 Adicionar sintetizador de texto</b>								

**ANEXOS | UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS | FACULTAD 4**  
**COMPONENTE PARA LA INTEGRACIÓN DE UN LECTOR DE PANTALLA AL MARCO DE TRABAJO XALIX**

EC 2.1	Presiona el botón “añadir sintetizador” en la ventana de configuración del sistema	V	V	N/A	N/A	V	Muestra una ventana en la que se pondrán los datos necesarios para adicionar el sintetizador, permitiendo aceptar o cancelar la operación.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 2.2	Cancela la operación.	N/A	N/A	N/A	N/A	V	Regresa a la página de configuraciones generales.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 2.3	Rellena los campos necesarios para adicionar un sintetizador y acepta la operación.	V	V	N/A	N/A	V	Muestra el mensaje “Los datos han sido guardados correctamente” y vuelve a la página de configuración general.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
		V	I	N/A	N/A	V	Muestra el mensaje “El sintetizador de texto que está tratando de adicionar aún no ha sido instalado, primero debe implementar las clases correspondientes al sintetizador que se desea agregar y colocarla en la carpeta TTS_DRIVER”.	
		I	I	N/A	N/A	V		
		I	V	N/A	N/A	V		
EC 2.4	Rellena los campos necesarios para adicionar un sintetizador y cancela la operación.	V	V	N/A	N/A	V	Regresa a la página de configuraciones generales.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.

**ANEXOS | UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS | FACULTAD 4**  
**COMPONENTE PARA LA INTEGRACIÓN DE UN LECTOR DE PANTALLA AL MARCO DE TRABAJO XALIX**

EC3 Editar datos del sintetizador								
EC 3.1	Selecciona editar sintetizador.	N/A	N/A	N/A	N/A	V	Muestra una ventana en la que se pondrán los datos que serán cambiados en el sintetizador, permitiendo aceptar o cancelar la operación.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 3.2	Cancela la operación.	N/A	N/A	N/A	N/A	V	Regresa a la página de configuraciones generales.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 3.3	Rellena los campos con los datos que serán cambiados del sintetizador y acepta la operación.	V	V	N/A	N/A	V	Muestra el mensaje “Los datos han sido guardados correctamente” y vuelve a la página de configuración general.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
		V	I	N/A	N/A	V	Muestra el mensaje “El sintetizado de texto que está tratando de adicionar aún no ha sido instalado, primero debe implementar las clases correspondientes al sintetizador que se desea agregar y colocarla en la carpeta TTSDriver”.	
		I	I	N/A	N/A	V		
EC 3.3		I	V	N/A	N/A	V		
EC 3.4	Rellena los campos con los datos que serán cambiados del sintetizador y cancela la operación.	V	V	N/A	N/A	V	Regresa a la página de configuraciones generales.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
SC4 Eliminar sintetizador								

**ANEXOS | UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS | FACULTAD 4**  
**COMPONENTE PARA LA INTEGRACIÓN DE UN LECTOR DE PANTALLA AL MARCO DE TRABAJO XALIX**

EC 4.1	Selecciona eliminar sintetizador.	N/A	N/A	N/A	N/A	V	Muestra el mensaje “Está seguro que desea eliminar el sintetizador seleccionado” y permite aceptar o cancelar la operación.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 4.2	Cancela la operación.	N/A	N/A	N/A	N/A	V	Regresa a la página de configuraciones generales.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 4.3	Acepta la operación.	N/A	N/A	N/A	N/A	V	Muestra el mensaje “El sintetizador ha sido eliminado correctamente” y vuelve a la página de configuración general.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
<b>EC 5 Adicionar voz</b>								
EC 5.1	Selecciona adicionar idioma.	N/A	N/A	N/A	N/A	V	Muestra una ventana en la que se pondrán los datos necesarios para adicionar la voz, permitiendo aceptar o cancelar la operación.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 5.2	Cancela la operación.	N/A	N/A	N/A	N/A	V	Regresa a la página de configuraciones generales.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.

**ANEXOS | UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS | FACULTAD 4**  
**COMPONENTE PARA LA INTEGRACIÓN DE UN LECTOR DE PANTALLA AL MARCO DE TRABAJO XALIX**

EC 5.3	Rellena los campos necesarios para adicionar un idioma y acepta la operación.	N/A	N/A	V	V	V	Muestra el mensaje “Los datos han sido guardados correctamente” y vuelve a la página de configuración general.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 5.4	Rellena los campos necesarios para adicionar un idioma y cancela la operación.	N/A	N/A	V	V	V	Regresa a la página de configuraciones generales.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
<b>SC6 Editar voz</b>								
EC 6.1	Selecciona editar voz.	N/A	N/A	N/A	N/A	V	Muestra una ventana en la que se pondrán los datos que serán cambiados en la voz, permitiendo aceptar o cancelar la operación.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 6.2	Cancela la operación.	N/A	N/A	N/A	N/A	V	Regresa a la página de configuraciones generales.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.

**ANEXOS | UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS | FACULTAD 4**  
**COMPONENTE PARA LA INTEGRACIÓN DE UN LECTOR DE PANTALLA AL MARCO DE TRABAJO XALIX**

EC 6.3	Rellena los campos con los datos que serán cambiados de la voz y acepta la operación.	N/A	N/A	V	V	V	Muestra el mensaje “Los datos han sido guardados correctamente” y vuelve a la página de configuración general.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 6.4	Rellena los campos con los datos que serán cambiados de la voz y cancela la operación.	N/A	N/A	V	V	V	Regresa a la página de configuraciones generales.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
<b>SC7 Eliminar voz</b>								
EC 7.1	Selecciona eliminar voz.	N/A	N/A	N/A	N/A	V	Muestra el mensaje “Está seguro que desea eliminar la voz seleccionada” y permite aceptar o cancelar la operación.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 7.2	Cancela la operación.	N/A	N/A	N/A	N/A	V	Regresa a la página de configuraciones generales.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.

EC 7.3	Acepta la operación.	N/A	N/A	N/A	N/A	V	Muestra el mensaje “La voz ha sido eliminada correctamente” y vuelve a la página de configuración general.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
--------	----------------------	-----	-----	-----	-----	---	--	---

**Tabla 17. Descripción del CP Descargar archivo de audio**

CP Descargar archivo de audio							
Descripción general							
Se inicia cuando el actor del sistema selecciona un texto para descargarlo como audio. Una vez descargado el archivo se finaliza.							
Condiciones de ejecución							
El usuario accede a un sitio web que tenga instalado el componente.							
El usuario define el texto a descargar.							
SC1 Descargar texto							
Escenario	Descripción	Etiquetas de role.	Textos alternativos de las	Información relevante del sitio web	Respuesta del sistema	Flujo central	
EC 1.1	Selecciona texto que se desea descargar.	N/A	N/A	N/A	Captura el texto seleccionado y permite descargarlo.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.	

**ANEXOS | UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS | FACULTAD 4**  
**COMPONENTE PARA LA INTEGRACIÓN DE UN LECTOR DE PANTALLA AL MARCO DE TRABAJO XALIX**

	No selecciona texto a descargar.	V	V	V	Permite descargar todo el contenido del sitio web.	
		V	V	I	Captura los datos existentes asociados a las variables con valores válidos y permite descargarlos.	
		V	I	V		
		I	V	V		
		I	V	I		
		I	I	V		
		I	I	I		
EC 1.2	Se procesa el texto, normalizándolo y sintetizándolo para convertirlo en un archivo de audio que podrá ser interpretado y reproducido dentro de un navegador web.	N/A	N/A	N/A	Se prepara para iniciar la descarga.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
EC 1.3	Selecciona la opción descargar archivo de audio.	N/A	N/A	N/A	Inicia la descarga.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
<b>SC2 Descargar texto mediante atajos del teclado</b>						

**ANEXOS | UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS | FACULTAD 4**  
**COMPONENTE PARA LA INTEGRACIÓN DE UN LECTOR DE PANTALLA AL MARCO DE TRABAJO XALIX**

EC 2.1	Presiona la combinación de teclas <b>Ctrl+</b> .	N/A	N/A	N/A	Si existe algún texto previamente seleccionado, el sistema procede a su descarga en archivo de audio en otro caso descarga un archivo de audio de todo el contenido del sitio web.	Se accede a <a href="http://localhost/XalixDev/web/app_dev.php/speech/">http://localhost/XalixDev/web/app_dev.php/speech/</a> , la cual es una página de ejemplo.
--------	--	-----	-----	-----	--	---