



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Proyecto Fin de Carrera

AraW²ord : una herramienta basada en
tecnologías Web para la escritura simultánea
de texto y pictogramas

Aarón Torrijos Bernal

Director: Francisco Javier Fabra Caro

Ingeniería Informática

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Noviembre 2013

AraW²ord: una herramienta basada en
tecnologías web para la escritura
simultánea de texto y pictogramas

RESUMEN

Los sistemas alternativos de comunicación surgen como complemento al lenguaje oral para ayudar a personas que padecen trastornos severos en el habla, bien de forma directa o bien de forma indirecta (lesiones o problemas neuronales, cognitivos, autismo, etc.). Estos sistemas facilitan la expresión sin utilizar la palabra, mediante la utilización de otros recursos.

En este contexto surge la herramienta AraWord, que permite la escritura simultánea de texto y pictogramas. Los pictogramas representan, de forma clara y sencilla, los conceptos más habituales para la comunicación cotidiana. AraWord es una herramienta open-source desarrollada por profesores y alumnos del Departamento de Informática de la Universidad de Zaragoza, y que en la actualidad cuenta con una amplia comunidad de usuarios a nivel internacional. AraWord se ofrece como un software de escritorio y en la actualidad no dispone de una interfaz orientada a servicios que pueda facilitar la integración de sus capacidades en entornos Web o dispositivos móviles.

En este proyecto se ha realizado el desarrollo de una versión Web de AraWord que expone todas las funcionalidades de la herramienta mediante servicios Web. La finalidad ha sido doble: por una parte, los servicios publicados pueden reutilizarse para desarrollar aplicaciones de propósito específico tanto para ordenadores de escritorio como para dispositivos móviles; por otra parte, se ha desarrollado una aplicación Web como caso de aplicación de estos servicios, llamada *AraW²ord*, accesible mediante cualquier dispositivo a través de un navegador.

Glosario

ARASAAC	Portal Aragonés de la Comunicación Aumentativa y Alternativa
AJAX	Asynchronous JavaScript and XML
BLOB	Binary Large Objects
CA	Comunicación Aumentativa
CAA	Comunicación Aumentativa y Alternativa
CATEDU	Centro Aragonés de Tecnologías para la Educación
CORS	Cross-origin resource sharing
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
LSE	Lengua de Signos Española
MEP	Message Exchange Patterns
REST	Representational State Transfer
SAAC	Sistemas Aumentativos y Alternativos de Comunicación
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SPC	Sistema Pictográfico de Comunicación
TIC	Tecnologías de la Información y Comunicación
WSDL	Web Services Description Language
XML	eXtensible Markup Language

Índice

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos del proyecto	2
1.3. Contenidos y alcance del documento	3
2. Conceptos previos	5
2.1. La comunicación Aumentativa y Alternativa	5
2.2. Sistemas Aumentativos y Alternativos de Comunicación (SAAC)	6
2.3. ARASAAC	6
2.4. Concepto de pictograma	7
2.5. AraWord	9
2.5.1. Concepto de pictograma en AraWord	9
2.5.2. La comunicación en AraWord	10
2.5.3. Puntos fuertes y débiles de AraWord	11
3. Proceso de análisis y desarrollo	13
3.1. Dedicación y distribución temporal de tareas	14
3.2. Análisis del problema: Diseño de la solución	14
3.2.1. Análisis de Requisitos y Casos de Uso	14
3.2.2. Arquitectura del sistema	16
3.2.3. Interfaz de servicios	17
3.2.4. Base de Datos	18
3.2.5. Modelo de proceso de desarrollo	19
3.3. Implementación: Desarrollo de la solución	19
3.3.1. Arquitectura del sistema y Funcionamiento	19
3.3.2. Implementación	20
3.3.3. Modelo de capas	26
3.3.4. Visión global de de interacción con el sistema	28
4. Resultados obtenidos	29
4.1. Interfaz de Servicios	29
4.2. Aplicación <i>AraW²ord</i>	31

5. Conclusiones y trabajo futuro	33
5.1. Conclusiones	33
5.2. Limitaciones	34
5.3. Trabajo futuro	35
5.4. Valoración personal	36
A. Evolución orientada a servicios	41
A.1. Diferencias entre aplicaciones de Escritorio y aplicaciones Web	41
A.2. Orientación a Servicios	42
A.3. Motor de los Servicios Web	44
B. Distribución temporal del proceso	47
B.1. Modelo Incremental	47
B.2. Distribución Temporal	48
B.2.1. Trabajo previo	48
B.2.2. Análisis del problema y diseño de la solución	48
B.2.3. Desarrollo de la solución	48
B.2.4. Verificación de la solución	49
B.2.5. Elaboración de documentación	49
C. Descripción detallada del proceso de desarrollo	51
C.1. Análisis de Requisitos	51
C.1.1. Requisitos funcionales	51
C.1.2. Requisitos No Funcionales	52
C.2. Casos de uso	52
C.3. Diseño de la aplicación	53
C.3.1. Diseño de la Interfaz de Servicios	53
C.3.2. Diseño de la base de datos	54
C.3.3. Diseño de la Aplicación	55
C.3.4. Servidor Web	56
C.4. Herramientas Utilizadas	57
C.5. Pruebas	58
D. Servicios Web Desarrollados	61
E. Manual de usuario	77
E.1. Manual de usuario	77
E.1.1. Página principal de la aplicación	77
E.1.2. Registro o identificación de usuarios	78
E.1.3. Registro de nuevo usuario	79
E.1.4. Opciones para usuarios registrados	79
E.1.5. Edición de texto	81
E.2. StoryBoard	83

Índice de figuras

2.1.	Pictograma que representa el concepto de 'perro'	7
2.2.	Pictogramas asociados a las palabras simples	8
2.3.	Pictograma asociado a la palabra compuesta	8
2.4.	Pictograma que representa el concepto de 'perro' en AraWord	9
2.5.	Ejemplo de texto escrito en AraWord	10
3.1.	Diagrama de Gantt	14
3.2.	Diagrama general de Casos de Uso de <i>AraW²ord</i>	16
3.3.	Arquitectura <i>AraW²ord</i>	17
3.4.	Servicio Web	17
3.5.	Arquitectura detallada	19
3.6.	<i>AraW²ord</i> usuario invitado	23
3.7.	<i>AraW²ord</i> usuario registrado	24
3.8.	Modelo de tres capas	26
3.9.	Proceso de interacción en <i>AraW²ord</i>	28
4.1.	Interfaz de Servicios	29
4.2.	WSDL del paquete UserTasks	30
4.3.	<i>AraW²ord</i>	31
A.1.	Capas del modelo SOA	43
A.2.	Apache Axis2	45
B.1.	Modelo Incremental	47
B.2.	Dedicación de tareas en porcentaje	50
C.1.	Diagrama de Casos de Uso de <i>AraW²ord</i>	52
C.2.	Diagrama de paquetes	53
C.3.	Base de Datos de <i>AraWord²</i>	55
C.4.	Diagrama de paquetes de la Aplicación <i>AraWord²</i>	56
C.5.	Pruebas con SoapUI	59
C.6.	Pruebas de integración	60
E.1.	Página de inicio de <i>AraW²ord</i>	77
E.2.	Ventana de login	78
E.3.	Ventana de registro de usuarios	79
E.4.	Pantalla de inicio para usuario registrados	79
E.5.	Cambio de preferencias de un usuario	80

ÍNDICE DE FIGURAS

E.6. Gestión de pictogramas	81
E.7. Cambio de nombre	82
E.8. Cambio de nombre realizado	82
E.9. Cambio de imagen	83

Capítulo 1

Introducción

Este proyecto ha sido realizado en la Universidad de Zaragoza, en el Departamento de Informática e Ingeniería de Sistemas (DIIS), en el cual han sido desarrollados otros proyectos relacionados con esta temática como son AraWord[1], TICO[2], TICO4Android[3], o AraBoard[4]. El objetivo principal es desarrollar un API¹ que implemente los servicios involucrados en AraWord mediante tecnologías Web. Posteriormente se realizará una aplicación basada en la utilización de dichos servicios.

1.1. Motivación

La comunicación aumentativa o CA es el término utilizado para definir el conjunto de formas, estrategias y métodos que se pueden utilizar para facilitar la comunicación a personas que presentan dificultades graves para ejecutar el habla. En la actualidad, se habla más de comunicación aumentativa y alternativa o CAA. Se habla de sistemas alternativos cuando se emplea un medio de comunicación diferente al habla y de sistemas aumentativos cuando se usan métodos que complementan al habla.

Las dificultades pueden surgir a cualquier edad y por diversos motivos entre los que destacan la parálisis cerebral, autismo, deficiencia mental, demencias, enfermedades degenerativas, etc.

Las Tecnologías de la Información y la Comunicación (TIC) juegan un papel muy importante para facilitar la comunicación. Se diseñan programas que potencian las capacidades comunicativas: el habla residual, el habla no funcional, los gestos, signos, pictogramas, etc. Actualmente existen herramientas que palían o minimizan estas deficiencias. A continuación se enumeran algunos de los ejemplos más populares:

- Comunicadores electrónicos: son sistemas alternativos que permite introducir mensajes en voz (grabada o sintética)
- Procesadores de texto y pictogramas: permiten el uso simultáneo de sistemas pictográficos y escritura.

¹Interfaz de programación de aplicaciones

1. Introducción

- Sistemas de predicción de escritura: utilizan la escritura como sistema aumentativo. Estos sistemas se utilizan cuando el aprendizaje está consolidado para evitar la lentitud asociada a la escritura que tienen las personas con un alto déficit motriz.

AraWord[1] es una aplicación de libre distribución, enmarcada dentro de una suite de herramientas para la comunicación aumentativa y alternativa, que consiste en un procesador de textos que permite la escritura simultánea de texto y pictogramas, facilitando la elaboración de materiales y adaptación de textos para las personas que presentan dificultades en el ámbito de la comunicación funcional.

AraWord se ha convertido en un proyecto de índole internacional contando con más de 350 descargas semanales en Sourceforge². Actualmente, AraWord está integrado en un proyecto más complejo, AraSuite³, que se compone de un conjunto de herramientas para la comunicación aumentativa y alternativa y cuenta con una media de 947 descargas semanales.

Esta aplicación es muy popular entre la comunidad de usuarios, pero en un mundo en el que la tecnología Web cada vez cobra más fuerza, AraWord cuenta con la restricción de poder usarse solamente en entornos de escritorio. Además con la aparición de nuevos dispositivos móviles, se hace necesaria la evolución de AraWord hacia la Web.

Actualmente, son muchas las aplicaciones que surgen cada día, muchas de ellas nativas para las diferentes plataformas pero muchas otras de carácter general. El desarrollo de una interfaz pública de los servicios utilizados en AraWord hará posible la constante evolución de la aplicación independientemente de la plataforma utilizada. Así mismo, el poder ejecutar la aplicación en un navegador abre las posibilidades y cubre un amplio rango de dispositivos en los que puede ser utilizada.

Hace un tiempo, la comunidad de usuarios y de desarrolladores pidió el poder usar este software en tablets y otro tipo de dispositivos, lo que motivó el desarrollo de este proyecto final de carrera.

1.2. Objetivos del proyecto

Con el desarrollo del presente proyecto lo que se consigue es crear una interfaz de servicios que implemente las principales funcionalidades presentes AraWord. Así mismo, también se plantea el desarrollo de una versión Web de dicho proyecto, *AraW²ord*, basada en estos servicios y así poder ejecutarla en cualquier dispositivo con navegador y conexión a Internet.

La principal ventaja de este desarrollo reside en la capacidad de adaptación a la evolución de la tecnología. Debido a que las operaciones que gestionan el negocio estarán publicadas y accesibles será fácilmente adaptable a una nueva tecnología. Desarrollar

² Sitio web de colaboración para proyectos de software, www.sourceforge.net

³<http://sourceforge.net/projects/arasuite/>

1. Introducción

una nueva versión supondría la codificación en el lenguaje deseado de la lógica de funcionamiento.

Puesto que no se trata de una migración del proyecto anterior sino de una versión con una filosofía totalmente distinta (orientación a servicios), se va a realizar una versión más ligera cuya funcionalidad será la misma en esencia (procesar textos con palabras y pictogramas).

A continuación se enumeran las principales funcionalidades para disponer de una visión global de la aplicación. La descripción detallada de dichos requisitos puede encontrarse de forma más detallada en el anexo C.

1. Posibilidad de editar texto con palabras simples y compuestas⁴
2. Conservación de la metodología establecida en ARASAAC[5] (distinción por colores según el tipo de palabra)
3. Posibilidad de incluir pictogramas propios
4. Posibilidad de enviar una frase por correo electrónico
5. Posibilidad de creación de perfiles de usuario
6. Utilización de verbos en todas sus formas verbales

Se dispondrán dos formas de interacción con la aplicación, una como usuario sin registrar y otra como usuario registrado (con esta última se tendrán más opciones como la carga de pictogramas propios o la edición de opciones, entre otras).

1.3. Contenidos y alcance del documento

El contenido de esta memoria se presenta en dos partes. En la primera se describe el trabajo realizado y las conclusiones obtenidas. La segunda parte consta de una serie de anexos que complementan la información y contienen información más técnica acerca de las tecnologías, metodologías, etc.

La parte principal se estructura en los siguientes capítulos:

- **Capítulo 1** - Introducción: Breve descripción del proyecto a realizar
- **Capítulo 2** - Conceptos previos: Explicación de los conceptos necesarios para la correcta comprensión de la memoria. Se explicarán los conceptos básicos de comunicación alternativa y aumentativa, el concepto de pictograma, etc.
- **Capítulo 3** - Proceso de análisis y desarrollo: Descripción de la metodología seguida para la realización del proyecto. Principales decisiones tomadas y justificación de las mismas.

⁴Se entiende que una palabra es compuesta cuando varias palabras se corresponden con una única realidad.(Ej. cabina de teléfono)

1. Introducción

- **Capítulo 4** - Resultados obtenidos: Breve descripción resultante del proceso de desarrollo.
- **Capítulo 5** - Conclusiones y trabajo futuro: Análisis de todo el proceso de realización del proyecto, así como del cumplimiento de objetivos. Descripción de fortalezas, debilidades, posibles mejoras y trabajo futuro.

Los anexos que se adjuntan se dividen en:

- **Anexo A** - Evolución orientada a servicios web
- **Anexo B** - Distribución temporal del proceso
- **Anexo C** - Descripción detallada del proceso de desarrollo
- **Anexo D** - Servicios Web desarrollados
- **Anexo E** - Manual de usuario

Capítulo 2

Conceptos previos

El objetivo de este capítulo es introducir los conceptos necesarios para la correcta comprensión del resto de la memoria. Se definen conceptos como comunicación alternativa, sistemas de comunicación alternativa, pictograma, etc. También se explica brevemente la implicación que tiene ARASAAC[5] con estos tipos de sistemas y su forma de trabajar con ellos.

2.1. La comunicación Aumentativa y Alternativa

La comunicación es un acto que consiste en la transmisión de información (mensaje) entre un emisor (elemento/persona que transmite la información) y un receptor (elemento/persona que recibe la información). Buena parte del proceso de aprendizaje se basa en la comunicación y no siempre es una tarea fácil para educadores y profesores. Personas afectadas por discapacidades físicas o psíquicas (sordera, parálisis cerebral, trastornos del espectro autista, entre otros) presentan graves dificultades en la comunicación oral y escrita. Generalmente dichos síntomas se manifiestan a edades tempranas y tienen repercusiones importantes en el proceso de aprendizaje.

Para ayudar a este colectivo de personas surge el término conocido como Comunicación Aumentativa y Alternativa (CAA) cuyo objetivo es presentar estrategias y métodos que permitan la comunicación entre personas con discapacidades específicas.

Los principales propósitos de la CAA son:

- Suministrar un medio de comunicación alternativo hasta que se restablezca la comunicación hablada de forma adecuada.
- Aportar un medio de comunicación alternativo de por vida, cuando no es posible o funcional la comunicación hablada del sujeto.
- Servir como medio de apoyo al desarrollo o al restablecimiento de la comunicación hablada.
- Servir como medio de apoyo en el aprendizaje de la lengua escrita.

2. Conceptos previos

2.2. Sistemas Aumentativos y Alternativos de Comunicación (SAAC)

Para dar soporte a la CAA surgen los Sistemas de Comunicación Aumentativa y Alternativa o SAAC. Éstos, son formas de expresión distintas al lenguaje hablado, que tienen como objetivo aumentar (aumentativos) y/o compensar (alternativos) las dificultades de comunicación y lenguaje de personas con discapacidad.

Los SAAC se clasifican en:

- **No asistidos:** También conocidos como SAAC sin ayuda, son aquellos en los que no es necesaria la utilización de ningún aparato o material. Los códigos utilizados no necesitan de ningún elemento externo al propio emisor. El ejemplo más característico sería el lenguaje de signos.
- **Asistidos:** Los SAAC asistidos o con ayuda se caracterizan porque los códigos que utilizan requieren un dispositivo externo. Ejemplos de estos sistemas serían los comunicadores electrónicos, los tableros de comunicación, etc.

En este proyecto se aborda el desarrollo de un SAAC asistido, por ello serán éstos sistemas en los que centraremos nuestra atención.

Existen múltiples productos de apoyo a la comunicación adaptados según las discapacidades presentes en las personas. Algunos cuentan con comunicadores de habla artificiales, otros incluyen salida de voz y la mayoría utilizan materiales visuales. Los recursos visuales pueden ser de diversos tipos, desde letras y palabras hasta fotografías, dibujos, pictogramas, etc.

Los sistemas pictográficos constituyen el recurso más extendido en personas que todavía no están alfabetizadas debido a la temprana edad o a alguna discapacidad. Éstos permiten un nivel de comunicación muy básico que se adapta bastante bien a personas con niveles cognitivos bajos. Los sistemas pictográficos más utilizados en la actualidad son el SPC¹ y el sistema pictográfico de ARASAAC²

2.3. ARASAAC

ARASAAC[5] es el Portal Aragonés de la Comunicación Aumentativa y Alternativa. Este proyecto forma parte del Plan de Actuaciones del Centro Aragonés de Tecnologías para la Educación (CATEDU). En ARASAAC se ofrecen recursos gráficos y materiales para ayudar a personas con trastornos severos en el habla. Todos los materiales se ofrecen bajo una licencia Creative Commons.

¹Sistema Pictográfico de Comunicación creado por Roxana Mayer en 1981 bajo licencia de tipo comercial

²Sistema pictográfico creado por el Centro Aragonés de Tecnologías para la Educación, profesionales del CPEE Alborada y Sergio Palao en 2008. Están accesibles bajo licencia Creative Commons en <http://www.catedu.es/arasaac/descargas.php>

2. Conceptos previos

Actualmente disponen de cinco catálogos con pictogramas en color y blanco y negro, fotografías, vídeos y fotografías en color en LSE³. Además cuenta con más de 13.000 pictogramas y éstos siguen aumentando. ARASAAC cuenta con una amplia comunidad y está presente en las principales redes sociales.

En este proyecto se utilizan los recursos pictográficos elaborados por ARASAAC y el CPEE Alborada. Dado que este sistema está muy difundido, y dado el carácter altruista del mismo, en este proyecto se decide usar dichos pictogramas, amén de que éstos mismos han sido los utilizados por los proyectos precursores.

2.4. Concepto de pictograma

Un pictograma es un signo que representa esquemáticamente un símbolo, un objeto real o una figura. Se trata de un diagrama que mediante imágenes o símbolos permite mostrar la realidad de forma que sea fácilmente comprensible y siguiendo un estándar.

Los pictogramas pueden ser utilizados como sistemas aumentativos o alternativos para representar una realidad concreta (ej: animal, persona), una realidad abstracta (ej: un sentimiento), una acción (ej: mirar) e incluso un elemento gramatical (ej: adjetivos, conjunciones).

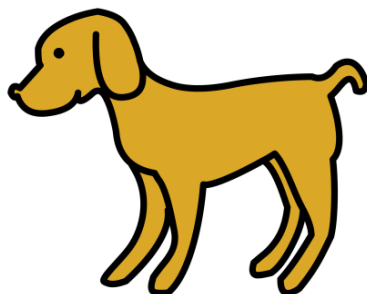


Figura 2.1: Pictograma que representa el concepto de 'perro'

Como se puede apreciar en la figura anterior, la realidad 'perro' es fácilmente comprensible mediante la imagen que representa dicha realidad. No obstante, existen palabras o grupos de palabras que puede representar realidades diferentes dependiendo de como estén agrupadas. Éste es el caso de las palabras compuestas⁴. Ejemplos de palabras compuestas son: centro comercial, cabina de teléfono, cepillo de dientes, etc.

A continuación se muestra un ejemplo de este tipo de palabras:

³Lengua de Signos Española

⁴Se dice que una palabra es compuesta si dos o más palabras se asocian con un único pictograma

2. Conceptos previos

- Pictogramas asociados a las palabras: cepillo, de, dientes:

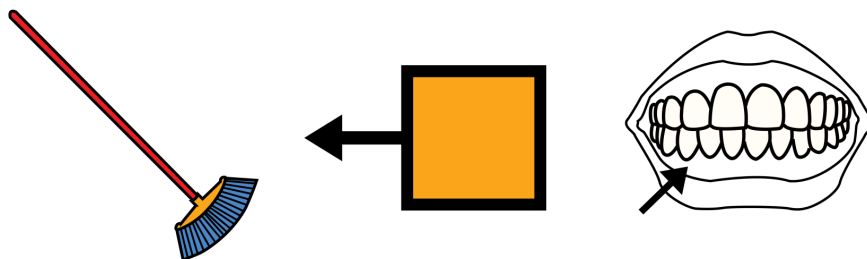


Figura 2.2: Pictogramas asociados a las palabras simples

- Pictograma asociado a la realidad: cepillo de dientes:



Figura 2.3: Pictograma asociado a la palabra compuesta

Cuando se produzca una combinación de palabras compuestas como la descrita en la Figura 2.2 el resultado debería ser un único pictograma (Figura 2.3)

Con el pictograma se identifica la realidad representada fácilmente pero en el proceso de aprendizaje es necesaria más información. Por ejemplo, es importante saber que tipo de palabra estamos visualizando (verbo, nombre común, etc.) y además puede ser interesante el poder ver la palabra que da nombre a la realidad representada. Por estas razones, en AraWord[1] se extendió el concepto de pictograma.

2. Conceptos previos

2.5. AraWord

AraWord[1] es un SAAC asistido que utiliza recursos pictográficos para llevar a cabo la comunicación. Consiste en un procesador de textos que permite la escritura simultánea de texto y pictogramas, facilitando la elaboración de materiales y adaptación de textos para las personas que presentan dificultades en el ámbito de la comunicación funcional.

AraWord se ejecuta como una aplicación de escritorio y para su funcionamiento incorpora una Base de Datos con Pictogramas elaborados por ARASAAC. Actualmente esta Base de Datos cuenta con más de 13000 pictogramas en siete idiomas diferentes.

Para la edición de texto, AraWord cuenta con múltiples opciones; se puede editar texto de la forma habitual, mediante la introducción de caracteres o también se da la posibilidad de copiar un texto externo al editor de textos y ver su representación pictográfica.

Permite la identificación de palabras compuestas, así como la composición o descomposición manual para formar uno o varios pictogramas respectivamente.

2.5.1. Concepto de pictograma en AraWord

En AraWord[1] el concepto de pictograma evolucionó para adecuarse a los requisitos pedidos por el CPEE Alborada. Un pictograma no sólo consta del signo o dibujo sino que también se almacenaba información del tipo de palabra así como la propia palabra.

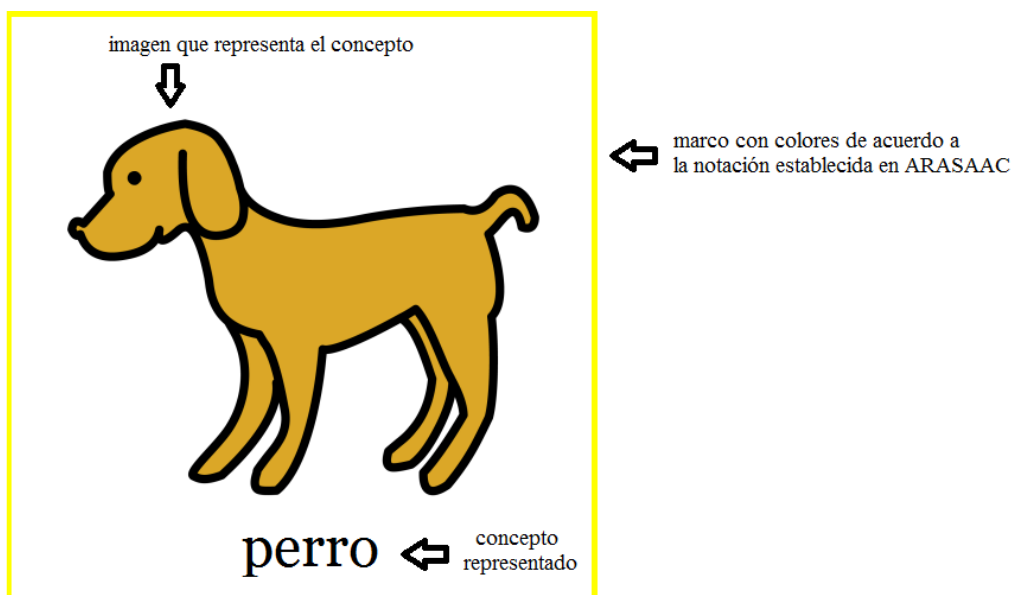


Figura 2.4: Pictograma que representa el concepto de 'perro' en AraWord

El concepto 'perro' es más completo, además de la representación gráfica se muestra

2. Conceptos previos

la palabra que da nombre a la realidad representada y se da información acerca del tipo de palabra. El tipo de palabra queda definido mediante un código de color. (ej: borde amarillo representa un nombre común).

En AraWord un término puede tener asociados varios pictogramas, lo que significa, que una misma realidad puede tener varias imágenes disponibles, el usuario puede elegir la que más se adecue al contexto en el que se encuentre (ej. diferentes imágenes para la realidad 'perro'). Así mismo, un pictograma puede tener asociado varios términos, esto es, diferentes realidades pueden representarse con la misma imagen (ej. varias razas de perros representados con la misma imagen).

2.5.2. La comunicación en AraWord

AraWord está disponible para varios Sistemas Operativos (Windows, Linux, MacOS). Para poder utilizarlo, hay que descargarse el software correspondiente de acuerdo con nuestro sistema operativo. Es un software que funciona para entornos de escritorio y tras unos sencillos pasos podemos instalar la aplicación. Al instalar la aplicación, ésta instala la base de datos con los pictogramas en nuestro equipo.

La comunicación con AraWord está basada en una hoja en blanco en la cual el logopeda o el alumno escriben texto y como respuesta a esa escritura se obtiene el pictograma asociado.

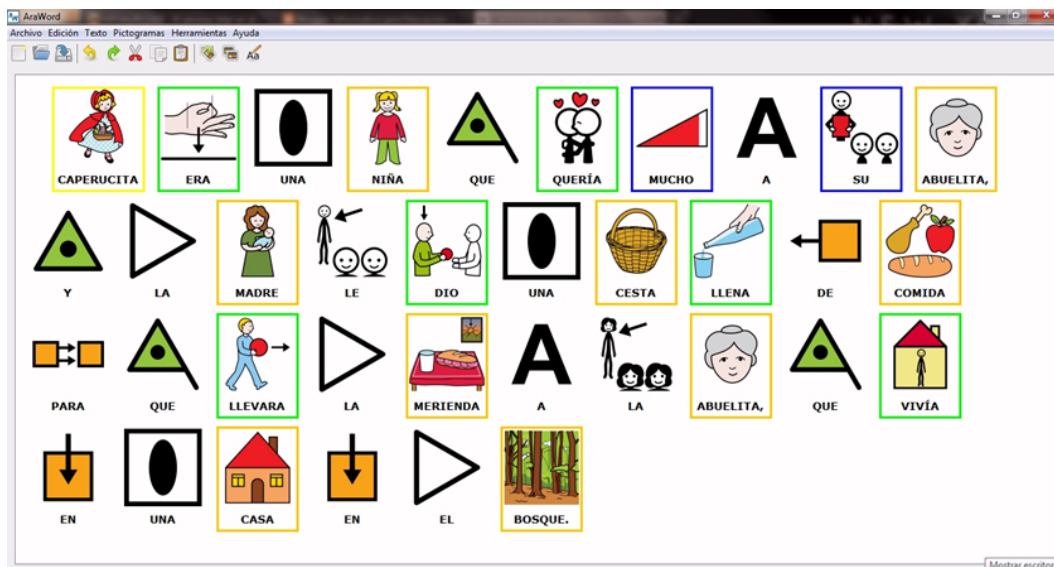


Figura 2.5: Ejemplo de texto escrito en AraWord

2. Conceptos previos

2.5.3. Puntos fuertes y débiles de AraWord

AraWord ha tenido un alto nivel de aceptación convirtiéndose en un proyecto de ámbito internacional. Lo avalan las más de 900 descargas semanales que se realizan a través Sourceforge de Arasuite, donde AraWord está integrado.

Por otra parte, el hecho de que AraWord solamente esté disponible como aplicación de escritorio limita su evolución y expansión. En la actualidad, gran parte de las aplicaciones están siendo orientadas hacia la Web. También cada vez son más los dispositivos nuevos que surgen con conexión a Internet, desde tablets o smartphones hasta pequeños miniordenadores como por ejemplo la Raspberry⁵. A pesar de la variedad de los sistemas operativos de estos dispositivos, en todos ellos se ofrece conectividad a Internet.

Con el desarrollo de una interfaz de los servicios ofrecidos en AraWord se va a poder extender AraWord a la Web y va a poder ser ofrecido en muchos más dispositivos.

⁵ Es una computadora completa en un sólo circuito. También conocidos como ordenadores de placa reducida(Single Board Computer o SBC)

Capítulo 3

Proceso de análisis y desarrollo

En este capítulo se presenta el análisis del problema a resolver. El problema reside en la realización de AraWord para poder ser ejecutado en un entorno Web. Se realizará un análisis de los requisitos, un análisis tecnológico de la solución y posteriormente se realizará la implementación del mismo. Para ello, se explicarán cada una de las fases en las que se ha dividido el proyecto y se abordarán las posibles dudas y problemas que surgieron, así como las soluciones y decisiones tomadas y el porqué de las mismas.

3. Proceso de análisis y desarrollo

3.1. Dedicación y distribución temporal de tareas

Este PFC comienza a finales de enero de 2013 y finaliza a principios de noviembre de 2013. Durante este periodo se pueden distinguir varias etapas. La Figura 3.1 muestra la distribución de tiempos mediante un diagrama de Gantt.

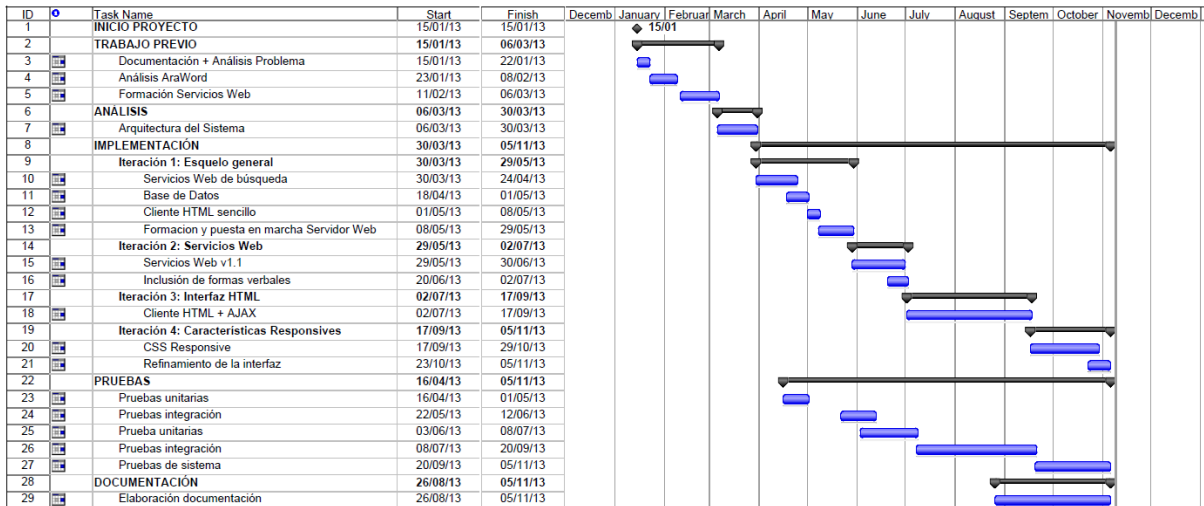


Figura 3.1: Diagrama de Gantt

Se pueden distinguir varias fases durante la realización del proyecto, las cuales son:

- Trabajo previo al desarrollo de la aplicación: En esta fase se hace una primera toma de contacto con lo que es la CAA y los SAACs. Se realiza la lectura de varios documentos relacionados con este tema.

Se realiza un análisis del proyecto previo “AraWord[1]: Un procesador de textos para comunicación aumentativa y adaptativa” realizado por Joaquín Pérez.

- Análisis del problema y diseño de la solución: En esta segunda fase se realiza el análisis del problema y se realiza el diseño de la solución.
- Desarrollo de la solución: Implementación de la solución diseñada.
- Verificación de la solución: Etapa en la que se realiza la verificación de la solución desarrollada.

3.2. Análisis del problema: Diseño de la solución

3.2.1. Análisis de Requisitos y Casos de Uso

La primera etapa del diseño de la solución pasa por la toma de requisitos. Para ello, se elaboran reuniones con el director de proyecto y con Joaquín Ezpeleta (director de proyectos previos a este y poseedor de gran conocimiento en el ámbito de la comunicación

3. Proceso de análisis y desarrollo

umentativa y alternativa). En dichas reuniones se establecen los objetivos y funcionalidades que debe cumplir la aplicación. A continuación de muestran de forma resumida los requisitos funcionales y no funcionales cuya descripción puede encontrarse de forma más detallada en el anexo C.

Requisitos Funcionales

- **RF-1** La aplicación podrá ser utilizada en dos modos distintos: usuario invitado y usuario registrado
- **RF-2** La aplicación permitirá la edición sencilla de texto con pictogramas asociados.
- **RF-3** La aplicación permitirá el registro de usuarios.
- **RF-4** Para un usuario registrado se podrá configurar el número de imágenes por línea, tamaño de letra, posición del texto, color del texto, el tamaño de la imagen, etc.
- **RF-5** Se permitirá la carga de imágenes propias para un usuario registrado
- **RF-6** Se permitirá editar/eliminar los pictogramas propios de usuarios registrados
- **RF-7** Se permitirá cambiar la imagen de un pictograma
- **RF-8** Existirán dos modos de edición de texto: inserción y edición.

Requisitos No Funcionales

- **RNF-1** La aplicación deberá funcionar correctamente en navegadores actuales que soporten HTML5, JavaScript y CSS3
- **RNF-2** La ejecución de la aplicación deberá ser independiente de la plataforma
- **RNF-3** El diseño de la aplicación está pensado para una resolución máxima de 960 píxeles de anchura.
- **RNF-4** Con resoluciones inferiores a 960 píxeles de anchura la interfaz se ajustará al tamaño de la pantalla del dispositivo
- **RNF-5** La sesión de un usuario permanecerá vigente hasta que se cierre el navegador o se cierre la sesión.
- **RNF-6** Solamente serán válidos los caracteres pertenecientes al conjunto $\{a-z|A-Z|0-9\}$ para la introducción de texto.
- **RNF-7** La aplicación estará liberada bajo una licencia GNU General Public License

3. Proceso de análisis y desarrollo

Los casos de uso son una técnica para especificar el comportamiento de un sistema. Con ellos se expresa como un agente externo a un sistema interactúa con él. En la figura 3.2 se ofrece una visión de la interacción del usuario con el sistema mediante los diagramas de casos de uso.

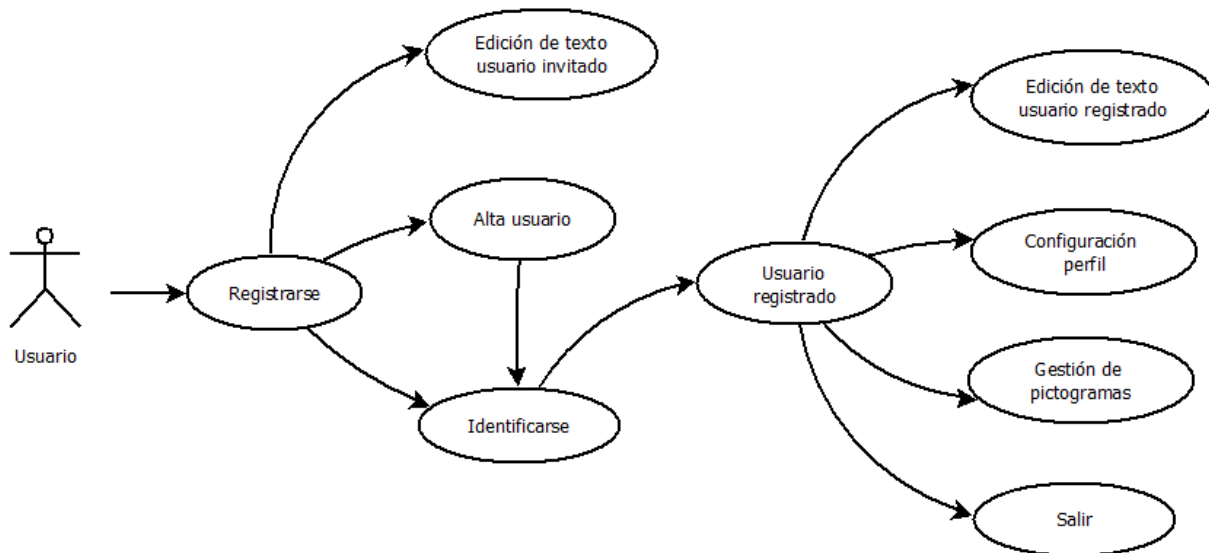


Figura 3.2: Diagrama general de Casos de Uso de *AraW²ord*

3.2.2. Arquitectura del sistema

En *AraW²ord* los usuarios no necesitan instalar ningún software adicional sino que sólo con disponer de un navegador y conexión a Internet es suficiente. Al tratarse de una aplicación Web, tanto la base de datos, como cualquier otro recurso residirá en Internet. A continuación se muestran los componentes que forman la arquitectura del modelo:

1. **Interfaz de Servicios** - Funciones que permiten la comunicación entre la interfaz de usuario, los pictogramas y la Base de Datos.
2. **Base de Datos** - Almacenamiento de diversos tipos de datos: referencias a los pictogramas, datos de usuario, etc.
3. **Servidor de recursos** - Contenedor con las imágenes de los pictogramas.
4. **Interfaz de usuario** - Frontal Web con la aplicación.

El principal problema recae en la comunicación entre los diferentes componentes. Desde una visión general, la arquitectura de AraWord puede verse de la siguiente forma:

3. Proceso de análisis y desarrollo

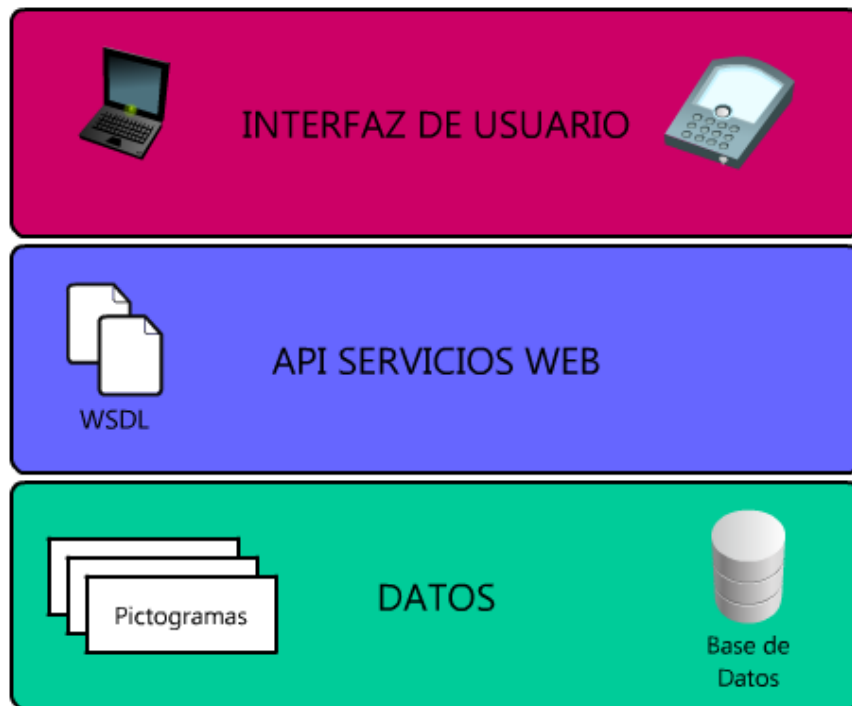


Figura 3.3: Arquitectura *AraW²ord*

En AraWord tanto la interfaz como la Base de Datos forman un todo. Sin embargo, en *AraW²ord* la aplicación (interfaz) está separada del resto de componentes y solamente se comunica a través de una interfaz de servicios.

3.2.3. Interfaz de servicios

Un servicio Web es el medio para exponer y hacer disponible la funcionalidad de los sistemas de información y de los servicios ofertados por una organización mediante las tecnologías de estándares Web. Básicamente, lo podríamos definir como una tecnología que sirve para intercambiar datos entre aplicaciones mediante un protocolo de comunicación.

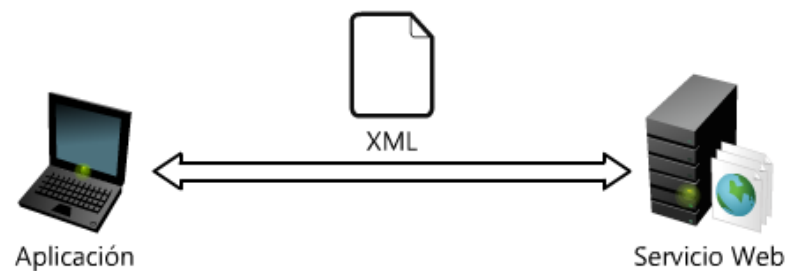


Figura 3.4: Servicio Web

3. Proceso de análisis y desarrollo

Se trata de una caja negra a la cual se le proporcionan unos datos de entrada y te devuelve un resultado.

Definición de los servicios

Para cubrir las funcionalidades básicas disponibles en AraWord se crea la Interfaz de Servicios. A continuación se describen de forma breve los principales servicios que conformarán el API¹:

- **Búsqueda de una palabra:** Es el principal servicio para que el procesador de textos funcione. Realiza la búsqueda de una palabra.
- **Añadir una imagen:** Servicio que permite cargar imágenes propias.
- **Edición de preferencias:** Servicio que permite a un usuario registrado cambiar sus preferencias. Entre otras, las preferencias que un usuario podrá definir son: número de pictogramas que se muestran en cada línea, tamaño y color de letra del texto, posición del texto(encima o debajo de la imagen), definir el número máximo de palabras que forman una palabra compuesta, tamaño de las imágenes, etc.
- **Gestión de imágenes propias:** Conjunto de servicios que permiten a un usuario ver qué imágenes tiene así como poder editarlas o borrarlas
- **Dar de alta un usuario:** Servicio que permite dar de alta a nuevos usuarios.
- **Verificación de usuario existente:** Servicio que verifica si el usuario que intenta acceder a la aplicación ha sido creado previamente.

3.2.4. Base de Datos

Es necesaria una Base de Datos que almacene información referente a los pictogramas, parámetros de usuarios, etc. En AraWord[1] se usó una SQLite 3.x². El motor SQLite se integra con el programa principal pasando a ser parte de él y no está pensado para realizar una gestión cliente-servidor por esto mismo.

Decidimos usar MySQL³ en su versión 5.5.32, que se trata de un gestor de bases de datos relacional, multihilo y multiusuario. Se mantienen las tablas presentes en AraWord y además se añaden otras nuevas que son necesarias. En el anexo C se describe de forma más detallada la base de datos y las tablas que la componen.

¹Interfaz de programación de aplicaciones

²SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (275 kiB)² biblioteca escrita en C. SQLite es un proyecto de dominio público¹ creado por D. Richard Hipp. A diferencia de los sistema de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo.

³MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario

3. Proceso de análisis y desarrollo

3.2.5. Modelo de proceso de desarrollo

Se elige un modelo incremental para el desarrollo del software. Éste modelo combina elementos del modelo lineal secuencial⁴ con la filosofía de construcción de prototipos. Cada secuencia lineal produce un 'incremento' del software hasta obtener el producto final.

3.3. Implementación: Desarrollo de la solución

En esta sección se describe el proceso de desarrollo basado en el análisis realizado en la sección anterior.

3.3.1. Arquitectura del sistema y Funcionamiento

El funcionamiento del sistema puede apreciarse en la figura 3.5. En ella se muestra la arquitectura de forma detallada y los procesos de comunicación entre los distintos componentes que la forman.

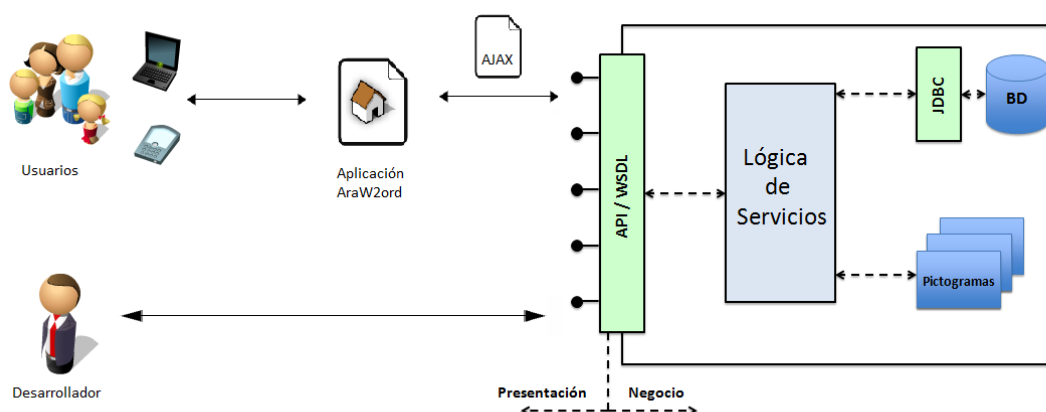


Figura 3.5: Arquitectura detallada

El funcionamiento es el siguiente:

- El usuario se conecta mediante el dispositivo a la Web de *AraW²ord*
- Cuando el usuario interactúa con la interfaz de *AraW²ord* se manda una petición HTTP a la interfaz de servicios con la operación deseada.
- El interfaz de servicios procesa la petición y se comunica con la Base de Datos o el servidor de Recursos si es necesario.
- El usuario recibe en la interfaz HTML el resultado devuelto por el interfaz de servicios

⁴El Modelo Lineal Secuencial sugiere un enfoque sistemático o más bien secuencial del desarrollo de software

3. Proceso de análisis y desarrollo

3.3.2. Implementación

Para el desarrollo de la aplicación se define la siguiente metodología de trabajo: Se establecen reuniones con el director de proyecto cada una o dos semanas dependiendo de la disponibilidad de ambos y de la carga de trabajo definida. En cada reunión se planifica una serie de hitos y al final de la misma se realiza un acta para llevar el seguimiento del proyecto.

El desarrollo de la aplicación se puede dividir en varias tareas:

1. **Servicios Web** - Desarrollo y pruebas de los servicios Web
2. **Base de Datos** - Integración de los servicios web con la base de datos
3. **Interfaz** - Desarrollo de la interfaz HTML y lógica de la aplicación
4. **Servidor Web** - Despliegue de la aplicación y de la interfaz de servicios en un servidor.

Servicios Web

Para el desarrollo de los servicios Web se decidió usar Eclipse⁵ como entorno de trabajo y Java como lenguaje de programación. El lenguaje de programación podía haber sido cualquier otro pero se decidió Java por comodidad y familiarización con el mismo. Para el desarrollo de los servicios Web nos hemos decantado por usar Apache Axis 2⁶ por su extendido uso, además Eclipse provee de una interfaz que permite generar y testar los servicios Web de forma relativamente sencilla.

El estándar utilizado ha sido SOAP⁷. SOAP es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de un intercambio de datos XML⁸. Este protocolo de intercambio de mensajes basado en XML define los siguientes elementos para realizar la comunicación:

- Sobre (envelope): Identifica el documento XML como un mensaje SOAP
- Cabecera (header): Contiene la información de la cabecera del mensaje
- Cuerpo (body): Contiene información de la llamada y respuesta
- Error: Elemento que contiene información acerca de si se ha producido algún error

La filosofía seguida para realizar los servicios Web ha sido Bottom-up. Consiste en un análisis ascendente en el que se parte de la implementación para posteriormente crear el Servicio Web. Ésta es una forma rápida de desarrollar los servicios Web.

A continuación se explica el funcionamiento de los servicios Web descritos en el análisis:

⁵<http://www.eclipse.org>

⁶<http://axis.apache.org/axis2/java/core/>

⁷Simple Access Protocol

⁸eXtensive Markup Language

3. Proceso de análisis y desarrollo

Servicio Web de búsqueda de palabras:

Proporcionado el nombre de la palabra, el servicio Web debe encontrar la imagen que se corresponde con dicha palabra en el servidor de pictogramas. Los pictogramas se encuentran alojados en el servidor de la Escuela de Ingeniería y Arquitectura (EINA). Se pueden encontrar en el dominio <http://www.gidhe.es/pictostorrijos/> Están distribuidos en nueve carpetas, así que para acceder a un pictograma en concreto es necesario proporcionar una url como la siguiente: <http://gidhe.es/pictostorrijos/1/10135.png>. El ejemplo anterior muestra el objeto 'horno', el cual está identificado por '10135', pero un usuario querrá escribir 'horno' y no '10135' para que se muestre el pictograma. Para la identificación del pictograma se emplea una base de datos que relaciona el identificador de la imagen (10135) con la palabra (horno).

El poder buscar formas verbales con sus conjugaciones es algo necesario (también demandado por el CPEE Alborada y presente en la versión escritorio). Se incluye una nueva tabla que contiene los pares [verbo,forma verbal] en español. En esta primera versión se incluye sólo este idioma dejando la inclusión de otros idiomas para un futuro próximo.

La búsqueda de palabras compuestas supone una importante degradación de las prestaciones, debido a esto se introduce en la función de búsqueda una funcionalidad que además de recuperar la palabra pedida da información de si puede formar parte de una palabra compuesta. Con esto ahorramos peticiones a los servicios Web en aquellos casos en que la palabra a buscar no pueda formar parte de una compuesta.

Servicio Web para añadir pictogramas:

Una funcionalidad necesaria es que un usuario pueda tener sus propios pictogramas, tal y como se podía hacer en AraWord[1]. A diferencia de una aplicación de escritorio, en una aplicación Web no puedes trabajar con rutas para el manejo de la selección de archivos. Por razones de seguridad los navegadores no permiten acceder al sistema de ficheros, así que para poder tener la imagen tuvimos que obtener su codificación base 64⁹ y almacenarla en la Base de Datos como un BLOB¹⁰.

⁹Base 64 es un tipo de codificación diseñado para que datos binarios sobrepasen capas de transporte que no son de 8-bits. La información codificada con este sistema dará como resultado una secuencia de letras, que constituye un archivo y puede ser decodificado y mostrado correctamente por cualquier navegador web.

¹⁰Binary Large Objects, objetos binarios grandes

3. Proceso de análisis y desarrollo

Servicio Web de edición de preferencias:

Permite cambiar las preferencias descritas en el análisis para el usuario que se ha identificado.

Servicio Web de gestión de imágenes propias:

La gestión de imágenes propias implica tres operaciones:

- Listado de imágenes: Muestra las imágenes propias del usuario identificado en la aplicación
- Edición de imágenes: Se permite cambiar el nombre.
- Eliminar imágenes: Se permite borrar una imagen propia.

Servicio Web de alta de usuario:

Permite dar de alta a un usuario en la aplicación a partir de su nombre, apellidos, email y contraseña. Para identificarse en la aplicación posteriormente su usuario será su email y su contraseña la misma que uso para el registro.

Servicio Web de verificación de usuario existente:

Comprueba que el nombre de usuario y la contraseña introducida existen en la aplicación y carga su perfil.

Base de Datos

Como se habló en la sección de diseño, para la Base de Datos se usa MySQL en su versión 5.5.32, que se trata de un gestor de bases de datos relacional, multihilo y multiusuario. Se crea la base de datos en el servidor Alkaid donde también residirá *AraW²ord*. Se mantienen las tablas presentes en AraWord y además se añaden otras nuevas que son necesarias. En el anexo C se describe de forma más detallada la base de datos y las tablas que la componen.

Interfaz

Para la realización de la interfaz se ha utilizado HTML¹¹ y CSS¹². Puesto que la aplicación debe funcionar correctamente en cualquier navegador y en cualquier resolución de pantalla se ha optado por usar un grid responsive.

Un grid es básicamente una hoja de estilos css. Hay infinidad de sistemas grid disponibles y la mayoría de ellos son gratuitos. Por su popularidad y versatilidad nos

¹¹HyperText Markup Language

¹²Cascading Style Sheets

3. Proceso de análisis y desarrollo

hemos decantado por usar bootstrap¹³. Bootstrap es el framework front-end desarrollado y liberado por Twitter. Implementa los nuevos estándares (HTML5 + CSS3) y funciona en cualquier navegador (IE 7/8/9, Firefox, Chrome, Safari, ...).

Este sistema está compuesto por 12 columnas con un ancho total de 960 píxeles. Además si no se es muy hábil con la maquetación css es una buena opción para realizar un diseño bastante aceptable. También es responsive, lo que significa que se adapta a cualquier resolución de pantalla.

El diseño de la interfaz es sencillo, intuitivo y minimalista. Dado que éste es un software que usará gente con discapacidades psíquicas y físicas se ha realizado el diseño teniendo esto en mente.

La aplicación consta de un menú con las opciones presentadas con pictogramas como botones. Se muestra el logo de AraWord cuya funcionalidad es presentar la página principal y un botón para entrar en la aplicación. La figura 3.6 refleja esta descripción.

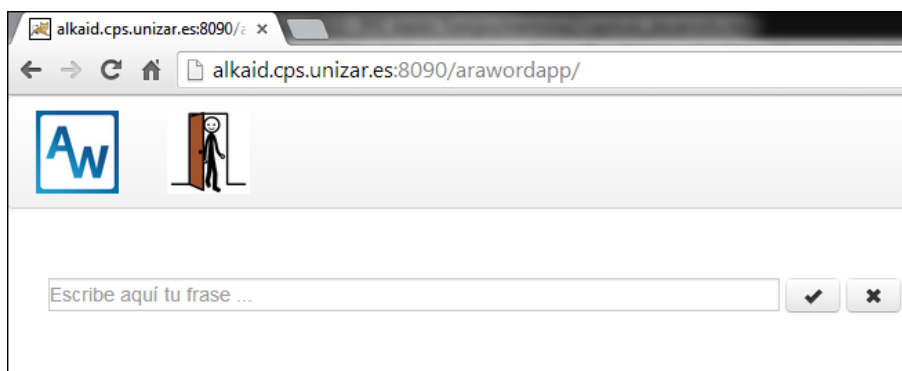


Figura 3.6: *AraW²ord* usuario invitado

Una vez registrado aparecen tres opciones más, una para configurar las preferencias de usuario, otra para la gestión de pictogramas y una última para cerrar la sesión.

¹³<http://getbootstrap.com/2.3.2/>

3. Proceso de análisis y desarrollo

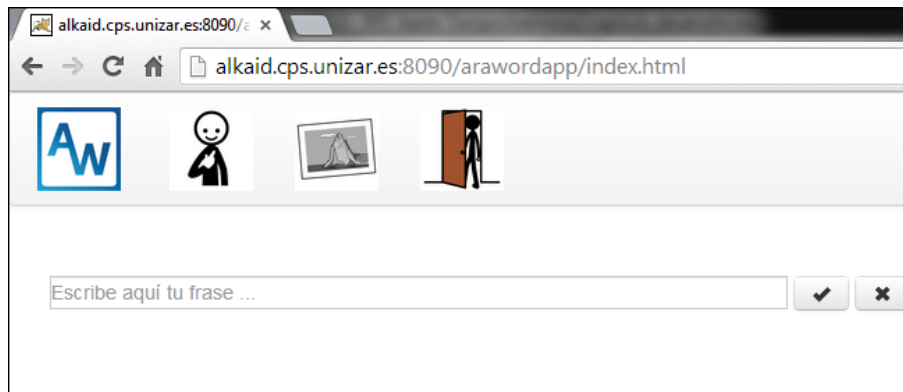


Figura 3.7: *AraW²ord* usuario registrado

En ambos modos (invitado o registrado) se presenta el cuadro de texto de edición dónde el usuario debe procesar el texto.

La interfaz HTML se gestiona a través de código JavaScript¹⁴ y jQuery¹⁵. Cada carácter que un usuario introduce en el cuadro de texto de la página principal se captura mediante un evento de teclado (keyup) y se procesa.

La introducción de ciertos caracteres especiales produce una llamada AJAX¹⁶ a la Interfaz de Servicios. Las peticiones Ajax son ejecutadas por código JavaScript, el cual envía una petición a una URL.

La URL a la cual se envía la petición es aquella en la que están publicados los Servicios. También es necesario comunicar la operación que queremos ejecutar y proporcionarle los parámetros necesarios para su ejecución en el servidor.

Para saber a qué método queremos hacer referencia y qué parámetros hemos de proporcionar se disponen de los ficheros de descripción de los Servicios Web o WSDL. En ellos, se describe la interfaz pública de los Servicios Web que contiene información del protocolo de comunicación y de los formatos de los mensajes a enviar.

El renderizado de los pictogramas en HTML se realiza una vez obtenida la respuesta del servidor. Para ello, se crea una tabla dinámica en HTML en la que se añaden los pictogramas (creados de forma dinámica con jQuery).

Funcionalidad de la Interfaz

Para dar funcionalidad al procesador de textos se debe poder ir tecleando palabras y que los pictogramas aparezcan de forma dinámica. Para conseguir esto se dispone de algu-

¹⁴Lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,3 basado en prototipos, imperativo, débilmente tipado y dinámico.

¹⁵ Biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML

¹⁶Asynchronous JavaScript and XML

3. Proceso de análisis y desarrollo

nas teclas especiales como el espacio (separador de palabras) y el backspace (para borrar caracteres). Las demás letras quedan disponibles para la introducción de texto, aunque no se permite poder usar todos los caracteres sino que se limita al subconjunto {a-z|A-Z|0-9}.

La unidad mínima de búsqueda se establece en una palabra. Se barajan dos opciones, hacer el tratamiento a nivel de carácter o de palabra y tras estudiar ambas opciones y medir sus rendimientos se decide que la unidad mínima de búsqueda sea una palabra. El tratamiento a nivel de carácter resulta muy costoso debido a la elevada cantidad de peticiones a los servicios Web.

Para el borrado se establece un time-out que dura hasta que se pulsa un carácter de edición {a-z|A-Z|0-9} o hasta que pasa un determinado tiempo. Con este time-out conseguimos realizar menos peticiones a los servicios Web lo que se traduce en un mejor tiempo de respuesta.

Palabras compuestas

Una palabra se dice que es compuesta si un único pictograma se corresponde con más de una palabra simple (ej: cepillo de dientes, centro comercial, patatas fritas, etc.).

Para encontrar este tipo de palabras es necesario comprobar si con las n-1 o n+1 palabras es posible formar una compuesta. Si estamos en modo edición normal, es decir con el cursor al final del texto sólo será necesario la comprobación con las n-1 palabras anteriores. En el caso que estemos editando una palabra en mitad del texto, ya sea añadiendo nuevos caracteres o eliminándolos será necesario comprobar las n-1 y n+1 palabras.

Esto supone realizar más peticiones al servicio web de búsqueda con la implicación de un mayor retardo. Después de varias pruebas, se asumió que el retardo obtenido en palabras compuestas con n=3 era asumible.

Arquitectura de despliegue: Servidor Web

Para poder tener la aplicación disponible y accesible al resto de mundo, es necesario desplegarla en un servidor Web. Se desarrolló un entorno en un servidor en el que se instaló un Tomcat¹⁷ para desplegar los servicios Web y un MySQL para la gestión de la base de datos.

Tomcat es un servidor Web con soporte de servlets¹⁸ y JSPs¹⁹ (métodos de creación de páginas Web dinámicas usando el lenguaje Java). A pesar de que Tomcat no es un servidor de aplicaciones como JBoss²⁰, es suficiente para nuestro propósito y más sencillo de configurar.

¹⁷<http://tomcat.apache.org/>

¹⁸Los servlets son objetos que corren dentro y fuera del contexto de un contenedor de servlets

¹⁹Java Server Pages

²⁰<http://www.jboss.org/>

3. Proceso de análisis y desarrollo

La jerarquía de directorios de Tomcat incluye un directorio llamado webapps que contiene las aplicaciones Web. Es en éste directorio donde se deben alojar los Servicios Web para tenerlos publicados al resto del mundo y es aquí también donde se aloja la aplicación *AraW²ord*.

3.3.3. Modelo de capas

Definidos Interfaz, Servicios Web y Base de Datos, podemos ver el sistema como un modelo de capas, en el que la Interfaz HTML es la capa de presentación, los Servicios Web constituyen la capa de negocio y la Base de Datos y los pictogramas se corresponden con el modelo de datos. La figura 3.8 muestra esta situación:



Figura 3.8: Modelo de tres capas

En la capa de presentación el usuario visualiza la aplicación, *AraW²ord*. Cuando el usuario interacciona con la aplicación ya sea introduciendo una palabra, identificándose,

3. Proceso de análisis y desarrollo

etc. se realiza una petición HTTP²¹ mediante AJAX²² a la capa de negocio.

La capa de negocio está formada por el Interfaz de Servicios, el cual procesa la petición y accede a la capa de datos. Para el acceso a la capa de datos se conecta el Interfaz de Servicios con MySql usando JDBC²³.

²¹Hypertext Transfer Protocol. Protocolo usado en cada transacción de la World Wide Web

²² Asynchronous JavaScript And XML

²³JDBC es un API de Java para acceder a manejadores de bases de datos

3. Proceso de análisis y desarrollo

3.3.4. Visión global de de interacción con el sistema

En la figura 3.9 se muestra el proceso llevado a cabo desde que el usuario realiza una acción hasta que visualiza el cambio que se ha producido

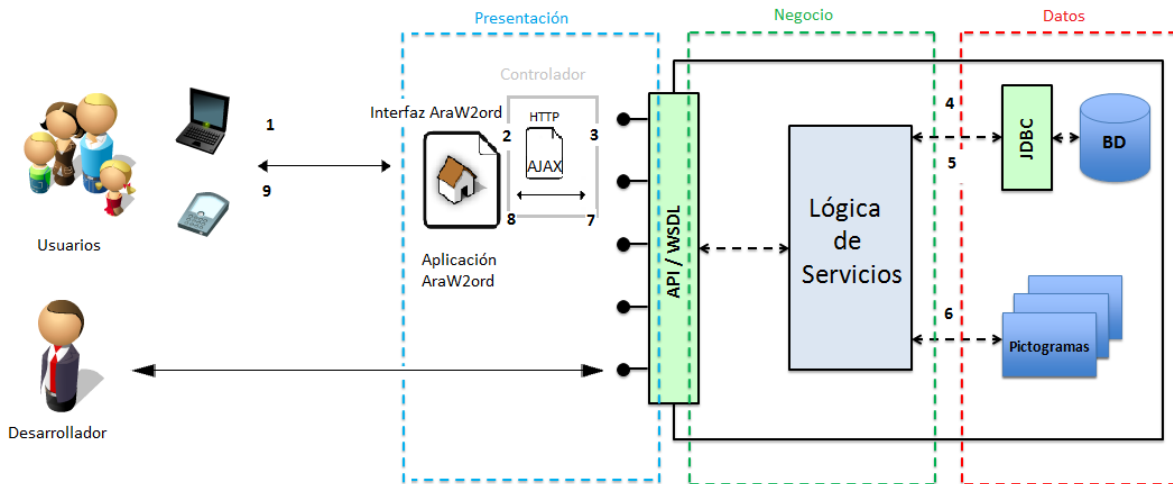


Figura 3.9: Proceso de interacción en AraW²ord

El proceso es el siguiente:

1. El usuario mediante el navegador accede a la url de AraW²ord y se muestra la aplicación.
2. Ante un cambio (introducción de una palabra, identificación, etc.) se llama a una función del Controlador JavaScript.
3. El Controlador realiza una petición AJAX mediante jQuery al Interfaz de Servicios.
4. El Interfaz de Servicios opera con la Base de Datos.
5. Se devuelven los datos de Base de Datos al Interfaz de Servicios.
6. Si la operación lo requiere (ej. búsqueda de palabra) se accede al Servidor de Pictogramas para recuperar la imagen que corresponda.
7. El Interfaz de Servicios tras procesar la petición devuelve el resultado al Controlador.
8. El Controlador presenta los datos en el Interfaz HTML.
9. Finalmente, el usuario puede ver el cambio que ha producido su acción.

Capítulo 4

Resultados obtenidos

A continuación se describe brevemente el resultado obtenido del desarrollo de la Interfaz de Servicios y de la aplicación.

4.1. Interfaz de Servicios

Como resultado de la implementación del Interfaz de Servicios, el resultado es la publicación de dichos servicios de forma pública y gratuita, de manera que estén accesibles al resto de desarrolladores.

En la siguiente url quedan publicados los Servicios Web:

`http://alkaid.cps.unizar.es:8090/arawordWebServices/services/listServices`

Se puede ver que están distribuidos por paquetes y accediendo a uno de esos paquetes podemos ver el WSDL¹ en el que se definen las operaciones disponibles. En la figura 4.1 se muestran los servicios disponibles y en la figura 4.2 un ejemplo de WSDL.

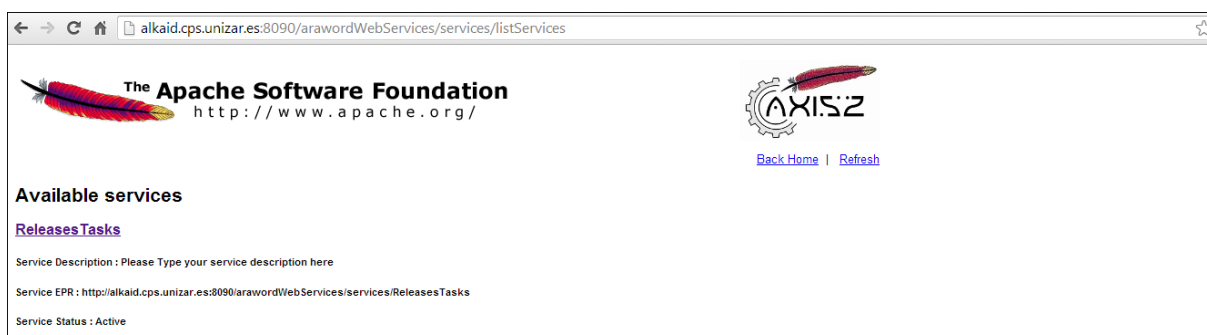



Figura 4.1: Interfaz de Servicios

¹Web Services Description Language, un formato XML que se utiliza para describir servicios Web

4. Resultados obtenidos



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:ns1="http://org.apache.axis2/xsd" xmlns:ns="http://users.org"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:ax21="http://general.org/xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" targetNamespace="http://users.org">
  <wsdl:documentation>Please Type your service description here</wsdl:documentation>
  <wsdl:types>
    <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://general.org/xsd">
      <xs:complexType name="Response">
        <xs:sequence>
          <xs:element minOccurs="0" name="description" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="error" nillable="true" type="ax21:ErrorWS"/>
          <xs:element minOccurs="0" name="status" nillable="true" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="ErrorWS">
        <xs:sequence>
          <xs:element minOccurs="0" name="code" type="xs:int"/>
          <xs:element minOccurs="0" name="description" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="operation" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="service" nillable="true" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="AddImageForUserDBResponse">
        <xs:complexContent>
          <xs:extension base="ax21:Response">
            <xs:sequence/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
      <xs:complexType name="RegisterNewUserResponse">
        <xs:complexContent>
          <xs:extension base="ax21:Response">
            <xs:sequence/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
      <xs:complexType name="SavePreferencesResponse">
        <xs:complexContent>
          <xs:extension base="ax21:Response">
            <xs:sequence/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:schema>
  </wsdl:types>

```

Figura 4.2: WSDL del paquete UserTasks

4. Resultados obtenidos

4.2. Aplicación *AraW²ord*

La aplicación está disponible en la url mostrada a continuación:
<http://alkaid.cps.unizar.es:8090/arawordapp/>

La figura 4.3 muestra el resultado de la aplicación en funcionamiento. Como se aprecia se ejecuta en un navegador Web y el menú principal está formado por dos o cuatro pictogramas, dependiendo si estamos identificados o no en la aplicación.

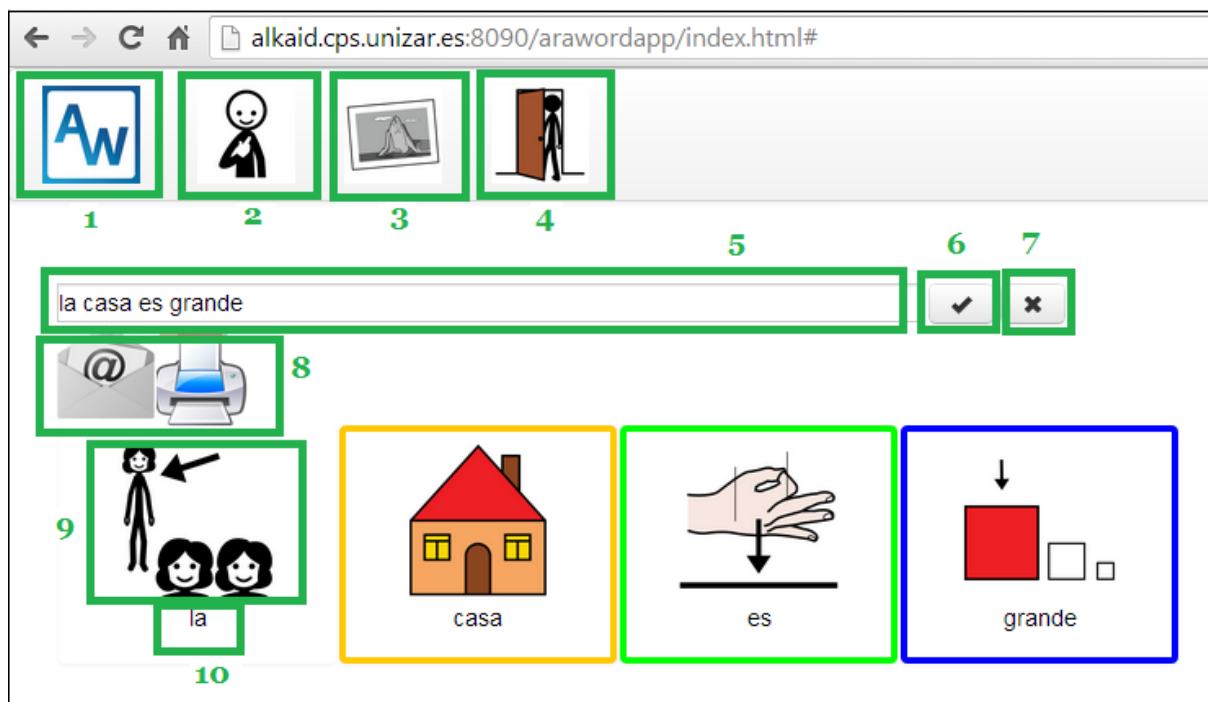


Figura 4.3: *AraW²ord*

Vamos a dar una breve descripción de los contenidos representados en la imagen:

- 1 - Logo de AraWord (lleva a la home, que es la pantalla mostrada)
- 2 - Permite la modificación de parámetros como el tamaño de letra (10), el tamaño de imagen (9), etc.
- 3 - Gestión de pictogramas propios
- 4 - Logout. Cierra la sesión actual
- 5 - Cuadro de introducción de texto
- 6 - Validación de la frase. Si no ha ocurrido ningún problema aparece la opción de mandar la frase por correo electrónico o de imprimirla (8)
- 7 - Borra la frase actual (resetea el cuadro de texto y borra todos los pictogramas)

4. Resultados obtenidos

- 8 - Opción para mandar la frase por correo electrónico o imprimirla
- 9 - Imagen del pictograma
- 10 - Palabra asociada a la imagen del pictograma

Para profundizar en el funcionamiento de cada opción y tener una idea más detallada de la aplicación se puede consultar el anexo E.

Capítulo 5

Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones obtenidas de la realización del proyecto. Se analiza el cumplimiento de objetivos, los principales problemas encontrados y se realiza un análisis de las posibles líneas de trabajo futuras. Por último se muestra una valoración personal de todo el trabajo realizado.

5.1. Conclusiones

En el presente trabajo se ha llevado a cabo el desarrollo de AraWord para poder utilizarse en un entorno Web. Para ello se propuso realizar primero una Interfaz de Servicios y posteriormente una aplicación Web basada en los Servicios desarrollados.

En general, se han cubierto todos los objetivos propuestos. El Interfaz de Servicios se ha realizado exitosamente y está publicado. En cuanto a la aplicación, aunque no presenta las mismas funcionalidades que la versión de escritorio debido a su imposibilidad o falta de sentido, sí ha mantenido la funcionalidad principal, que es la forma de procesar el texto. Las opciones de copiado, borrado y pegado (disponibles en AraWord) no están disponibles en esta versión. El motivo es que el uso de esta aplicación va a ser diferente que la versión de escritorio, puesto que estas opciones se usan sobre todo para copiar textos de gran extensión como puede ser un cuento y esto carece de sentido en un dispositivo de pocas pulgadas como puede ser un tablet o un smarphone.

Por lo demás, esta primera versión Web es bastante completa. Se ha mantenido el poder incluir pictogramas propios y además se ha incluido nuevas funcionalidades, como el poder enviar una frase por correo y la posibilidad de crear perfiles de usuarios.

5. Conclusiones y trabajo futuro

5.2. Limitaciones

El presente trabajo cuenta con una serie de limitaciones que son necesarias tener en cuenta.

- **Navegadores:** En el desarrollo se ha introducido código en HTML5¹ y en jQuery². La aplicación funcionará correctamente en aquellos navegadores que soporten dichas tecnologías
- **Subida de imágenes:** La inclusión de imágenes propietarias está permitida usando la aplicación desde un ordenador pero no desde un dispositivo móvil (tablets o smartphones). Esto es así debido a que ciertos sistemas operativos no permiten la subida de ficheros por motivos de seguridad. No obstante, en el dispositivo móvil un usuario podrá ser capaz de modificar las imágenes y/o eliminarlas.

A lo largo de la realización del proyecto han surgido diversos problemas. A continuación se describen los más relevantes:

Montar la arquitectura fue una de las tareas más costosas, puesto que hay que tener muy claro cómo se realiza la comunicación entre todos los componentes y hay que saber configurar diversos ficheros. Además se montaron tres entornos diferentes: un entorno local, otro en OpenShift³ y el último y definitivo en un Servidor. Un problema adicional fue la aparición de un error de CORS⁴. CORS es un mecanismo que permite utilizar JavaScript en una página web para hacer peticiones XMLHttpRequest a otro dominio. Este cruce de dominios no está permitido en los navegadores web utilizados en la actualidad por razones de políticas de seguridad. Finalmente, se optó por alojar tanto los servicios Web como la aplicación bajo el mismo dominio.

Un problema que no hemos sido capaces de solucionar ha sido el tratamiento de la letra “ñ”. Para el tratamiento de las letras, se trabaja con su codificación en ASCII⁵. En los teclados de los PCs funciona correctamente ya que se corresponde con el código ASCII 192, pero en algunos dispositivos (smartphones o tablets) el funcionamiento no es el esperado ya que no existe un código para la ñ, al menos uno que pueda capturar JavaScript. Cabe decir, que en los dispositivos más populares como son la tablet de Apple o la de Samsung así como los iPhone y Samsung Galaxy entre otros, la ñ devuelve un código 0. Este 0 indica un valor no definido que nosotros hemos atribuido a la ñ ya que no vamos a tener ningún otro valor no definido puesto que restringimos los valores al subconjunto {a-z|A-Z|0-9}

La introducción de verbos ha dado bastantes problemas por cuestiones de codificaciones (a pesar de permitir tildes en la codificación éstas no se guardaban correctamente en Base de Datos por lo que se ha optado por quitar las tildes en esta primera versión.

¹HyperText Markup Language, versión 5

²Fast, small, and feature-rich JavaScript library, www.jquery.com

³<https://www.openshift.com/>

⁴Cross-origin resource sharing

⁵American Standard Code for Information Interchange

5. Conclusiones y trabajo futuro

Además se dieron problemas al realizar las inserciones debido a la gran cantidad de datos, del orden de 801,023 inserciones.

También se debe mencionar que *AraW²ord* no cuenta con todas y cada una de las funcionalidades presentes en AraWord. Las funcionalidades que no están presentes en esta versión son el copiado y pegado de texto y la posibilidad de deshacer y rehacer las últimas acciones realizadas. Esto es así debido a que estas opciones carecen de sentido en un entorno Web. Por el contrario se incluyen otras nuevas como poder enviar la frase escrita por correo electrónico o la creación de perfiles de usuario.

Aunque en algunas ocasiones los problemas descritos desembocaron en algún retraso respecto a la planificación establecida, éstos fueron asumibles y no supusieron mayores contratiempos en el desarrollo de la aplicación

5.3. Trabajo futuro

La versión actual podría considerarse una primera versión estable de la versión Web de AraWord, no obstante hay ciertas mejoras que podrían tenerse en cuenta y que no se han podido realizar por falta de tiempo y por exceder el objetivo del proyecto, puesto que ésta es una primera versión. A continuación se hace un resumen de las posibles mejoras y cambios que podrían llevarse a cabo y de las líneas futuras:

- **Optimizar el tiempo de respuesta de los servicios Web:** Ésta es la tarea más difícil en una aplicación Web que usa servicios Web. El problema que tiene es que no es algo que dependa únicamente de la eficiencia del código desarrollado, puesto que también depende de la velocidad de la red en la que se esté usando. Se podría estudiar alguna forma de realizar menos llamadas a los servicios web.
- **Implementar funciones de borrado, copiado y pegado:** Aunque no se han tenido en cuenta por razones de uso en dispositivos con pantallas de tamaño reducido, la aplicación también puede utilizar en un PC con pantalla de mayores dimensiones. En este caso dichas funcionalidades cobrarían más sentido.
- **Diseño gráfico para dispositivos de resoluciones superiores a 960 píxeles:** Teniendo en cuenta la opción anterior, si se extiende el uso de esta aplicación a los PCs se podría modificar el diseño, puesto que esta basado en un grid de 960 píxeles. se esta desaprovechando el espacio en pantallas de mayor resolución. La idea sería aumentar las características responsives del diseño para cubrir estas resoluciones.
- **Múltiples perfiles:** Se ha establecido un perfil por usuario aunque cabe la posibilidad de desarrollar varios perfiles que facilitaría a profesores y educadores poder trabajar con diferentes tipos de alumnos.
- **Audio de la frase:** Desarrollar la opción que permita reproducir el audio de una frase.
- **Verbos multilinguaje:** Inclusión de más idiomas para las conjugaciones de los verbos en la base de datos. De momento sólo están disponibles los verbos en español.

5. Conclusiones y trabajo futuro

- **JSONP:** Modificar los Servicios Web para devolver los datos en formato JSON y utilizar JSONP para poder realizar peticiones entre diferentes dominios. Esto permitiría poder tener alojados los servicios Web y la aplicación en servidores diferentes.
- **API con servicios Web en REST:** Implementar los servicios Web con tecnología REST.

5.4. Valoración personal

Llegados al punto de la elección de un PFC todavía no tenía claro el ámbito en el que enfocaría los últimos meses de mi carrera. Después de haber vivido la experiencia de realizar prácticas en empresa y de plantearme la posibilidad de realizar el PFC en el ámbito empresarial me dí cuenta que aquello no era lo que estaba buscando.

Lo que buscaba era poder desarrollar algo útil y cuyo único fin no fuera el de poder rentabilizarlo. Así que este proyecto se adecuaba de forma perfecta a mis necesidades.

El impacto social del proyecto era una realidad presente, pues se trataba de realizar una aplicación que ayudara a la educación de personas con discapacidades físicas y/o psíquicas y problemas en la comunicación. Además la Escuela de Ingeniería y Arquitectura (EINA), más en concreto, el Departamento de Informática e Ingeniería de Sistemas (DIIS) ya había colaborado en otras ocasiones con otros desarrollos para dicho colectivo de personas.

Una vez concluida la realización del PFC puedo decir que estoy satisfecho con el trabajo realizado, aunque se podría haber alargado mucho más. Cada vez que se terminaban unos requisitos iban surgiendo nuevas ideas y nuevas opciones que se podrían haber incluido. Hay que decir que el desarrollo Web da múltiples posibilidades, como ejemplo, decir que pesamos en incluir *AraW²ord* en las redes sociales, el poder postear una frase en Facebook o Twitter nos parecía interesante, aunque al final no se incluyó porque excedía el planteamiento inicial en cuanto a tiempo y esfuerzo.

Los objetivos iniciales se han cubierto y aunque ésta no es la versión definitiva de lo que puede llegar a ser *AraW²ord* si que es la primera versión estable y completa.

Desde un punto de vista más técnico, la principal dificultad de este proyecto radicaba en la orientación a servicios. Durante la carrera, las aplicaciones que se desarrollan suelen ser en entornos de escritorio. El poder conocer el funcionamiento de una aplicación Web, el aprender a poner una aplicación Web en funcionamiento, con todo lo que ello supone, configuración de servidores, Bases de Datos, etc. ha sido de gran valor para mí.

Por otra parte, también he profundizado en el desarrollo Web con la realización de la aplicación. He aprendido los principios básicos de la programación Web, utilizando lenguajes como HTML, JavaScript, hojas de estilo CSS, aspectos que durante la carrera apenas se ven.

5. Conclusiones y trabajo futuro

Finalmente, debo destacar lo mucho que he podido aprender de mi director, que desde un punto de vista de gestión de proyectos, me ha enseñado la importancia de fijar metas realistas, gestionar el tiempo de forma apropiada y aprender a resolver problemas o simplificarlos para poder abordarlos de mejor manera.

Bibliografía

- [1] EZPELETA MATEO, JOAQUÍN y PÉREZ MARCO, JOAQUÍN *AraWord : un procesador de textos para comunicación aumentativa y adaptativa, 2011*
- [2] EZPELETA MATEO, JOAQUÍN y PALACIO JULIÁN, CAROLINA, *TICO e 1.0: mejora y extensión de la aplicación TICO para su primera distribución estable, 2010*
- [3] EZPELETA MATEO, JOAQUÍN, MARCO RUBIO, JAVIER y *TICO4Android: Implementación de TICO para dispositivos móviles basados en Android, 2013*
- [4] BALDASSARRI SANTA LUCÍA, SANDRA y FERRER DOMINGO, EDUARDO, GARCÍA AZPIROZ, MARTA, *Creación de herramientas software de apoyo a la comunicación alternativa y aumentativa, 2012*
- [5] ARASAAC, PORTAL ARAGONÉS DE LA COMUNICACIÓN AUMENTATIVA Y ALTERNATIVA, *<http://www.catedu.es/arasaac/>*
- [6] TALLER DE SERVICIOS WEB Y SOA, *Máster de Bases de Datos e Internet, Curso 2011/2012*
- [7] PROYECTO TICO, *<http://www.proyectotico.es/wiki/index.php/AraWord>*

Anexo A.

Evolución orientada a servicios

En este anexo se describe el proceso necesario que supone el cambio de una aplicación de escritorio a una aplicación Web basada en una filosofía orientada a servicios. Para ello, se analizan las principales diferencias entre plataformas en las que se ejecutan y posteriormente se describen los aspectos básicos a tener en cuenta y las decisiones que se deben tomar.

A.1. Diferencias entre aplicaciones de Escritorio y aplicaciones Web

Para tener claro que aspectos se deben implementar y qué es necesario cambiar hay que tener en cuenta las principales diferencias entre el desarrollo de aplicaciones de escritorio y aplicaciones Web. Así mismo también conviene realizar una comparativa de ventajas y desventajas que influyen para realizar una aplicación Web o no.

Una **aplicación de Escritorio** es aquella que está instalada en el equipo del usuario y es ejecutada directamente por el sistema operativo. Su rendimiento depende de los elementos hardware del equipo como son memorias RAM, discos duros, etc.

Las principales ventajas y desventajas son:

Ventajas	Desventajas
Su ejecución no requiere comunicación con el exterior	Acceso limitado al equipo donde están instaladas
El tiempo de respuesta es muy rápido	Dependen del SO ¹ del equipo
Suelen ser bastante seguras	Requieren instalación y actualización personalizada

A. Evolución orientada a servicios

Una **aplicación Web** está instalada en un Servidor y su ejecución requiere disponer de un navegador Web y conexión a Internet.

Las principales ventajas y desventajas son:

Ventajas	Desventajas
Portabilidad: se ejecutan desde cualquier dispositivo	Es necesaria una conexión a Internet
Idóneas para aplicaciones multiusuario	Dependencia de una buena conexión a Internet
Suelen ser aplicaciones ligeras	El Servidor debe tener las prestaciones necesarias
Consumen pocos recursos del equipo	Tiempos de respuesta más lentos
Fáciles de actualizar y mantener	Compatibilidad con diferentes tipos de navegadores
Independientes del SO	

Una vez analizadas las principales diferencias podemos hacernos a la idea de los componentes necesarios para el desarrollo de una aplicación Web:

- Dispositivo con conexión a Internet
- Servidor donde alojar la aplicación

A.2. Orientación a Servicios

La arquitectura orientada a servicios o SOA² es un software de diseño y un patrón de diseño software basado en piezas discretas de software para proporcionar funcionalidad como servicios a otras aplicaciones.

Un servicio es una unidad autónoma de funcionalidad con una interfaz estable y pública que puede ser invocado por un cliente.

La idea para lograr esto consiste en desarrollar una serie de módulos o paquetes que estarán alojados en el Servidor y accesibles mediante una interfaz pública. En la figura A.1 se pueden diferenciar las distintas capas que componen el modelo SOA.

²Service-oriented architecture

A. Evolución orientada a servicios

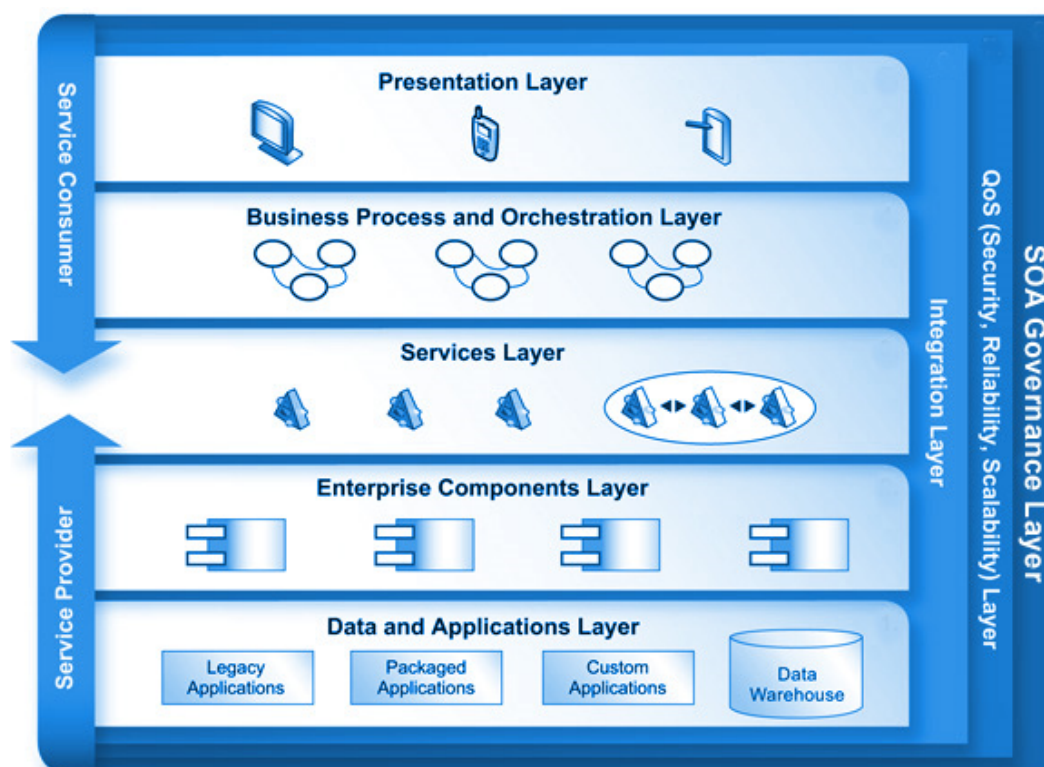


Figura A.1: Capas del modelo SOA

Para desarrollar la capa de Servicios actualmente existen dos arquitecturas: SOAP³ y REST⁴.

SOAP es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambios de datos en XML. Las operaciones se definen como puertos WSDL⁵. Estos ficheros describen todas las funciones de la interfaz así como el tipo de datos de entrada y de salida.

REST no está considerado como una arquitectura propiamente dicha. Es un estilo de arquitectura software que se centra en el uso de estándares para la transmisión de datos sin necesidad de contar con una capa adicional. Las operaciones se solicitan mediante GET, POST, PUT y DELETE lo que supone que no es necesaria ninguna implementación especial para consumir estos servicios.

En REST los servicios no publican un conjunto de métodos sino que lo que se publican son recursos. Un recurso se puede considerar como una entidad que representa un concepto de negocio que puede ser accedido públicamente.

Para este proyecto hemos elegido SOAP ya que es un estándar y es una arquitectura estable, aunque REST cada vez cobra más fuerza ya que se basa en los principios que hicieron posible la Web.

³Simple Object Access Protocol

⁴Representational State Transfer

⁵Web Services Description Language

A. Evolución orientada a servicios

A.3. Motor de los Servicios Web

El motor usado para el desarrollo de los servicios Web ha sido Apache Axis2⁶. Axis2 proporciona un completo modelo de objetos y una arquitectura modular que facilita la tarea de añadir funcionalidad y dar soporte a nuevos servicios Web.

Las principales características de Axis2 son:

- Axis2 soporta SOAP 1.1 y SOAP 1.2 y además integra soporte para REST.
- Velocidad
- Uso reducido de memoria
- Despliegue instantáneo
- Servicios web asíncronos
- Soporte de MEP⁷
- Flexibilidad
- Estabilidad
- Despliegue orientado a componentes
- Framework de transporte
- Soporte de WSDL
- Agregados
- Composición y extensibilidad

La figura A.2 muestra el motor Apache Axis2 desplegado en el Servidor Tomcat:

⁶<http://axis.apache.org/axis2/java/core/>

⁷Message Exchange Patterns

A. Evolución orientada a servicios

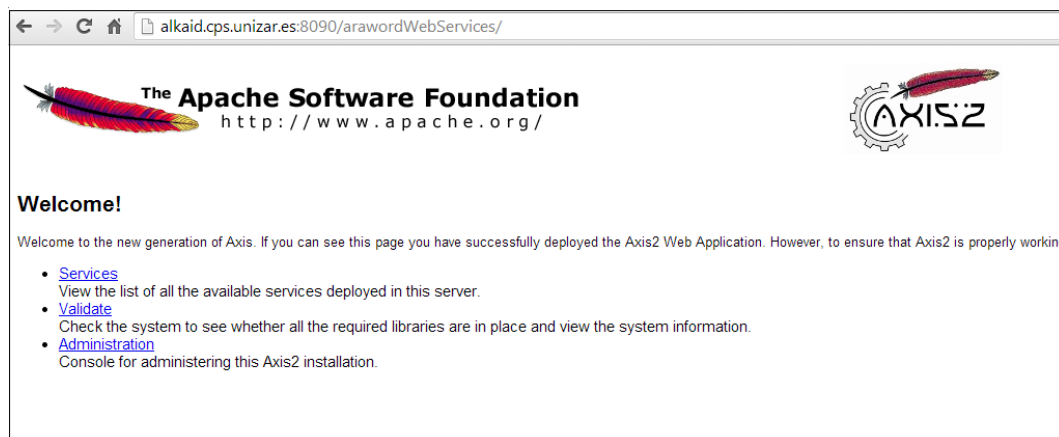


Figura A.2: Apache Axis2

Con el Interfaz de Servicios correctamente publicado y desplegado en nuestro Servidor Web ya disponemos de la lógica de negocio para nuestra aplicación. La Interfaz de la aplicación Web residirá en los diversos clientes HTML desplegados en el Servidor Web, con lo que nuestra aplicación queda accesible al resto del mundo.

A. Evolución orientada a servicios

Anexo B.

Distribución temporal del proceso

B.1. Modelo Incremental

Para la realización de este proyecto se ha elegido un modelo de proceso incremental. Éste combina elementos del Modelo Lineal Secuencial con la filosofía interactiva de Construcción de Prototipos. Como se muestra en la Figura B.1, el modelo incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software. El primer incremento generalmente es un producto esencial denominado núcleo.

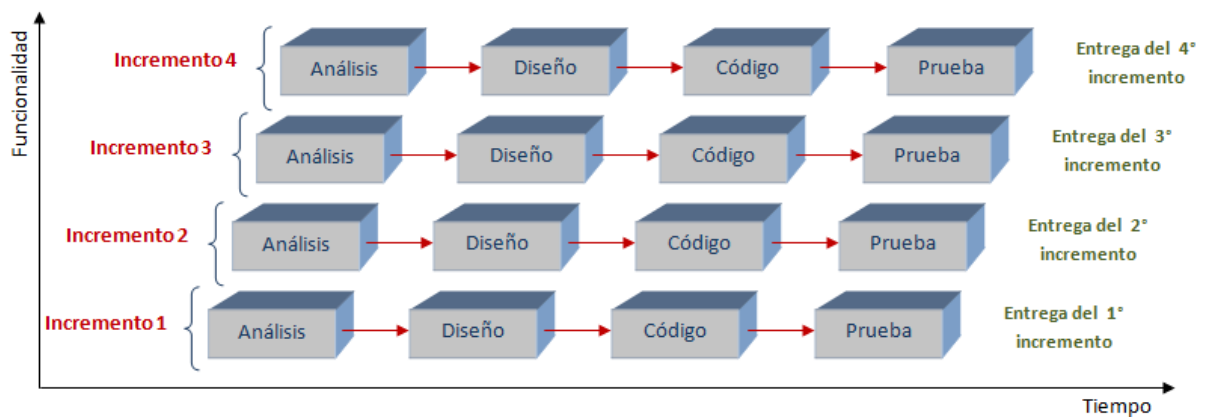


Figura B.1: Modelo Incremental

Como primer incremento podemos definir el primer prototipo de la aplicación compuesto por un cliente sencillo HTML con un cuadro texto y el servicio Web de búsqueda de palabras. A partir de ahí, se van definiendo los demás incrementos hasta desarrollar el sistema completo.

B. Distribución temporal del proceso

B.2. Distribución Temporal

En el capítulo de análisis y desarrollo se mencionó que se podían diferenciar las fases:

- Trabajo previo al desarrollo de la aplicación
- Análisis del problema y diseño de la solución
- Desarrollo de la solución: Implementación de la solución diseñada.
- Verificación de la solución

B.2.1. Trabajo previo

La primera fase tiene lugar entre finales de Enero y principios de Marzo. La primera tarea consiste en una búsqueda de información y la lectura de documentos relacionados con la comunicación aumentativa y alternativa.

Después de tener un conocimiento general del ámbito en el que se va a trabajar, se analiza el proyecto previo, AraWord[1]. Se realiza el estudio de dicho proyecto mediante la descarga del software y código de Sourceforge.

En esta fase también se realiza una formación tecnológica en Servicios Web, realizando lecturas y prácticas, ya que esta es la tecnología que se usará en buena parte del desarrollo del proyecto. Así mismo también se realiza una pequeña formación en diseño Web para el desarrollo de la aplicación.

B.2.2. Análisis del problema y diseño de la solución

El análisis del problema tiene lugar en Marzo. En esta fase se analizan las necesidades de los usuarios y se realiza el análisis de requisitos. La toma de requisitos se realiza mediante reuniones con el director de proyecto y con Joaquín Ezpeleta, que fue director del proyecto AraWord y tiene una estrecha relación con el CPEE Alborada.

Se diseña la arquitectura de la solución. Para ello se definen todos los componentes de los que constará el proyecto (Base de Datos, Servidor Web, etc) y se especifica cómo debe ser el funcionamiento básico.

B.2.3. Desarrollo de la solución

La fase de desarrollo tiene lugar de finales Marzo a principios de Noviembre. Durante esta fase podemos definir varias etapas o incrementos. Siguiendo la filosofía del modelo incremental, la idea en esta fase es ir desarrollando incrementos que sean totalmente funcionales. En esta fase también es necesaria la realización de algunos periodos de formación para la programación Web (HTML5, CSS3, JavaScript, etc.)

Podemos definir los siguientes incrementos:

B. Distribución temporal del proceso

1. Implementación del esqueleto general de la aplicación. En este primer incremento es uno de los más costosos, puesto que se realiza el desarrollo de la primera versión de la Interfaz de Servicios. Esta primera versión solamente se compone de la función de búsqueda de palabras, pero para su elaboración es necesaria también la configuración del Servidor Web y de la Base de Datos, además de la realización de un interfaz HTML en el que poder ejecutar la aplicación.
2. En un segundo incremento se diseñan el resto de funciones que conforman la Interfaz de Servicios.
3. En la tercera iteración se realiza el interfaz HTML con el resto de opciones que utilizan el resto de servicios Web (opciones de login y registro de usuarios, subida de imágenes propias, etc.)
4. En el cuarto incremento se añaden las características responsives para que la aplicación pueda adaptarse a cualquier tamaño de pantalla y se mejora la interfaz de usuario.

B.2.4. Verificación de la solución

La verificación tiene lugar durante el desarrollo de la solución. Tan pronto como hay desarrollado un módulo se realizan las pruebas unitarias del mismo y cuando se integra con los demás se realizan las correspondientes pruebas de integración.

Además, una vez terminada la aplicación se realizan pruebas globales en diferentes dispositivos.

B.2.5. Elaboración de documentación

Durante la implementación y las pruebas también se realiza la elaboración de la presente memoria.

En la Tabla B.1 y en la figura B.2 se muestra el total de horas invertidas y el porcentaje dedicado a cada tarea:

Tarea	Horas
Formación	80
Análisis y Diseño	120
Implementación	189
Pruebas	93
Documentación	85
Total	567

Tabla B.1: Dedicación de tareas en horas

B. Distribución temporal del proceso

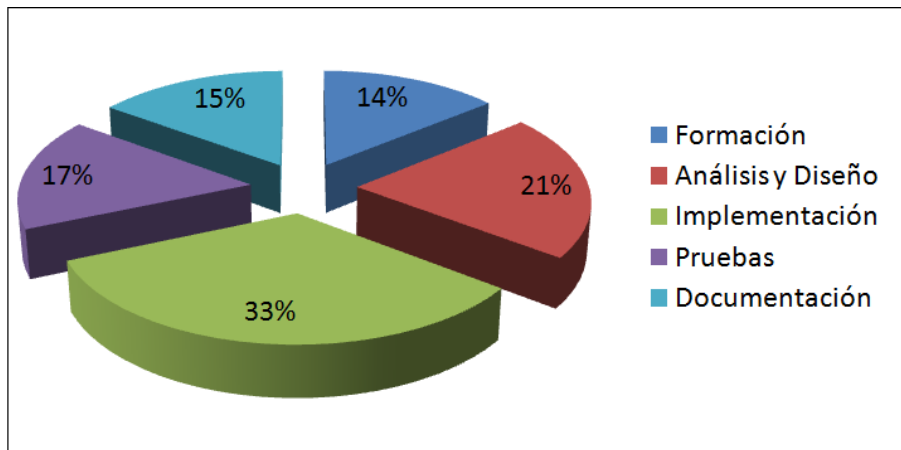


Figura B.2: Dedicación de tareas en porcentaje

Anexo C.

Descripción detallada del proceso de desarrollo

Este anexo complementa al capítulo 3 y describe de forma más detallada y técnica el proceso de desarrollo del software. Se describen las fases de análisis, diseño, implementación y pruebas.

C.1. Análisis de Requisitos

La primera etapa del desarrollo comienza por definir los requisitos que la aplicación debe cumplir. La toma de requisitos se lleva a cabo a partir de los requisitos definidos en AraWord[1] y en reuniones con el director de proyecto.

Algunos requisitos han ido cambiando a lo largo del desarrollo del proyecto. A continuación se muestran los requisitos tal y como quedaron en su versión final divididos en requisitos funcionales y no funcionales.

C.1.1. Requisitos funcionales

- RF-1:** La aplicación debe permitir la edición sencilla de texto con sus pictogramas asociados
- RF-2:** La aplicación permitirá el registro de usuarios.
- RF-3:** La aplicación debe poder ser utilizada en dos modos distintos: usuario invitado y usuario registrado
- RF-4:** Para un usuario registrado se deberá poder configurar el número de imágenes por línea, tamaño de letra, posición del texto, color del texto, el tamaño de la imagen
- RF-5:** Se debe permitir la carga de imágenes propias para un usuario registrado
- RF-6:** Se permitirá editar/eliminar los pictogramas propios de usuarios registrados
- RF-7:** Se permitirá cambiar la imagen de un pictograma
- RF-8:** Deberá ser posible cambiar el texto de un pictograma

C. Descripción detallada del proceso de desarrollo

C.1.2. Requisitos No Funcionales

- RNF-1:** La ejecución de la aplicación deberá ser independiente de la plataforma
- RNF-2:** La aplicación deberá funcionar correctamente en navegadores actuales que soporten HTML5, JavaScript y CSS3
- RNF-3:** El diseño de la aplicación está pensado para una resolución máxima de 960 píxeles de anchura
- RNF-4:** Con resoluciones inferiores a 960 píxeles de anchura la interfaz se ajustará al tamaño de la pantalla del dispositivo
- RNF-5:** La sesión de un usuario permanecerá vigente hasta que se cierre el navegador o se cierre la sesión
- RNF-6:** Solamente serán válidos los caracteres pertenecientes al conjunto $\{a - z|A - Z|0 - 9\}$ para la introducción de texto
- RNF-7:** La carga de imágenes no estará disponible en dispositivos móviles o tablets
- RNF-8:** La Interfaz de Servicios Web estará accesible mediante una url y será de acceso público
- RNF-9:** La aplicación estará liberada bajo una licencia GNU General Public License

C.2. Casos de uso

Los diagramas de casos de uso describen los pasos que deben realizarse para llevar a cabo un proceso. Un caso de uso es una secuencia de interacciones entre un sistema y un agente externo al sistema.

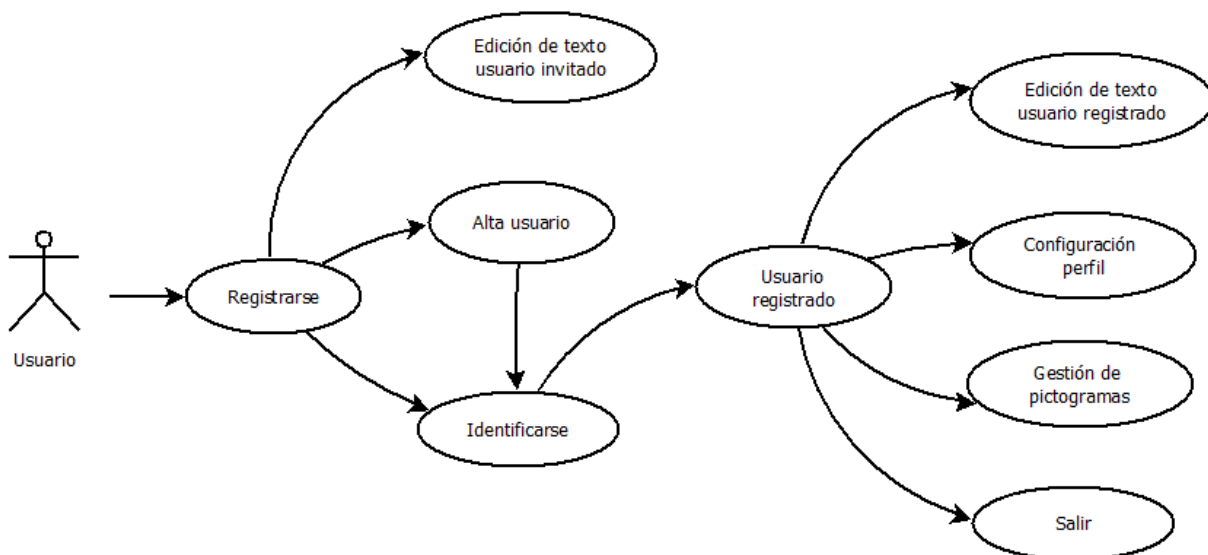


Figura C.1: Diagrama de Casos de Uso de *AraW²ord*

La figura C.1 muestra el diagrama de casos de uso modelado para *AraW²ord*. Los principales actores o usuarios de la aplicación son profesores o alumnos.

C. Descripción detallada del proceso de desarrollo

C.3. Diseño de la aplicación

El diseño final de la aplicación se compone de dos partes, por un lado el diseño de la Interfaz de Servicios y por otro el de la aplicación propiamente dicha. Por último se describe el diseño de la arquitectura, en el que se describe como se realiza el despliegue de la Interfaz de Servicios y de la aplicación en un Servidor Web.

C.3.1. Diseño de la Interfaz de Servicios

El diseño de la Interfaz de Servicios se muestra en la figura C.2 mediante la división en subsistemas o paquetes. En éstos se muestra el sistema dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones.

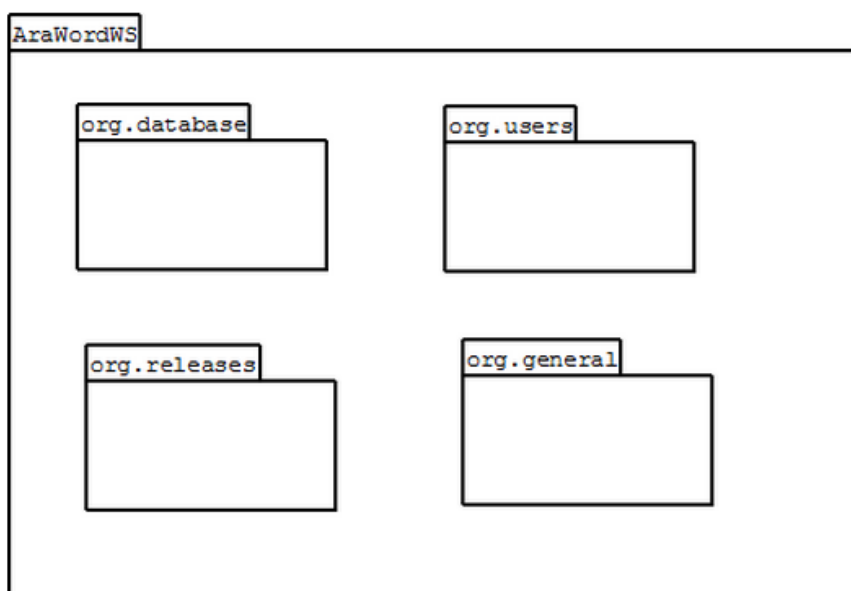


Figura C.2: Diagrama de paquetes

La aplicación se compone de los siguientes paquetes:

- org.database:** Contiene las clases para gestionar el acceso e interacción con la base de datos
- org.users:** Contiene las clases para gestionar las operaciones propias de usuarios
- org.releases:** Contiene las clases para gestionar las operaciones propias de versiones
- org.general:** Contiene clases de carácter general, estructuras de datos o herramientas necesarias.

Además de las clases mencionadas, para el correcto funcionamiento de la Interfaz de Servicios se ha requerido el uso de las siguientes bibliotecas externas:

C. Descripción detallada del proceso de desarrollo

- **org.apache.commons.codec.binary.Base64:** Biblioteca para codificar y decodificar imágenes en base 64.
- **mysql-connector-java-5.1.23:** Driver JDBC de Java para MySQL.

C.3.2. Diseño de la base de datos

Para el diseño de la base de datos se ha partido del modelo existente en AraWord[1] y se ha extendido. En AraWord la base de datos consta de tres tablas y están gestionadas con SQLite. A diferencia de los sistema de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica.

En este proyecto se ha optado por tener un servidor de bases de datos MySQL versión 5.5.32 dado que la base de datos no se integra con el programa sino que se accederá a ella por medio de los servicios web. Además se han añadido más tablas, las cuales se definen a continuación:

- **main:** Tabla presente en AraWord[1]. Esta tabla está formada por 5-tuplas que contienen toda la información referente a un pictograma. Las tuplas están formadas por estos 5 términos:

word: Nombre del concepto

idL: Clave del lenguaje de la palabra

idT: Clave del tipo de la palabra

name: Nombre normalizado de la imagen guardado en el servidor de pictogramas

nameNN: Nombre no normalizado de la imagen. No utilizado, se mantiene por compatibilidad con TICO.

- **language:** Tabla presente en AraWord. Contiene los pares (id, lenguaje).
- **type:** Tabla presente en AraWord. Contiene los pares (id, tipo de palabra).
- **main_for_user:** Esta tabla es nueva. Se ha añadido para guardar las imágenes propias de cada usuario. Consta de 6-tuplas con los valores [word, idL, idT, name, nameNN, idUser]. Los primeros cinco valores son los mismos que los definidos en la tabla main y el último valor hace referencia al usuario que posee dicho pictograma.
- **encoded_images_bytes:** Tabla nueva. Guarda la codificación de las imágenes contenidas en la tabla main_for_user.
- **users:** Tabla que contiene la información y las preferencias de un usuario.
- **verbs:** Tabla que contiene los verbos en español. Contiene los pares (verbo, forma verbal).
- **track:** Tabla nueva. Tabla que guarda el registro de las frases. Se utilizará en un futuro para poder sacar estadísticas de imágenes más usadas, etc.

C. Descripción detallada del proceso de desarrollo

- **track_info:** Tabla nueva. Describe la información presente en la tabla track.
- **releases:** Tabla nueva. Permite gestionar versiones de la base de datos. No usada en esta versión.
- **application:** Tabla que contiene datos referentes a la versión actual de la aplicación.

Las tablas application y releases no se usan en esta versión, pero se plantean para poder utilizarlas en un futuro si se extiende a TICO la orientación a Servicios.

A continuación se muestra el diagrama de la base de datos:

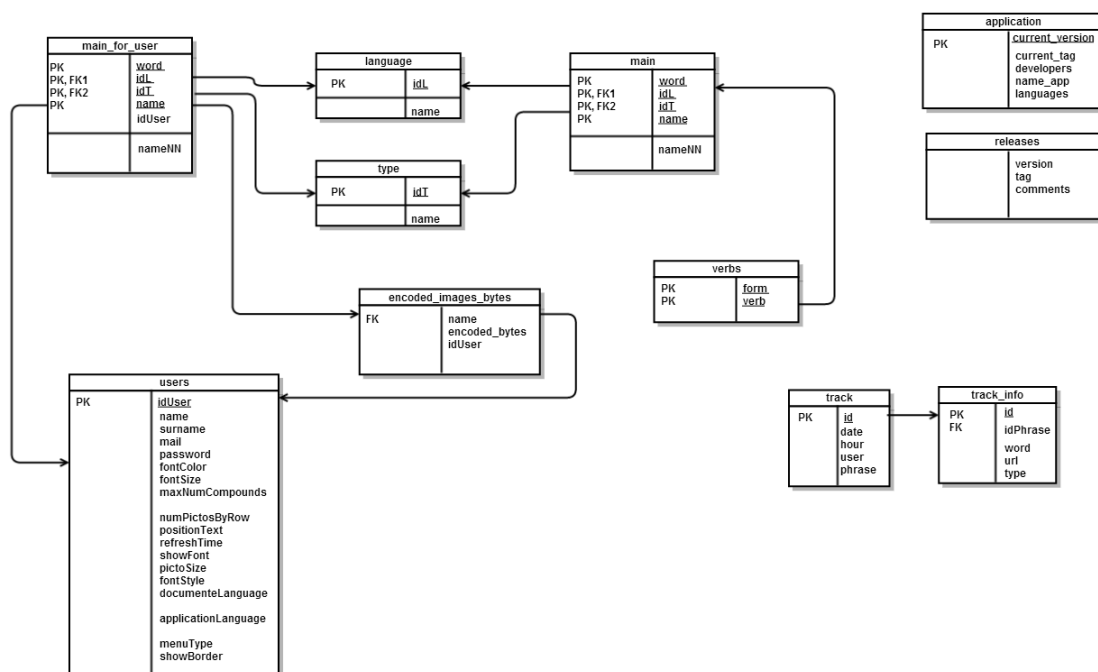


Figura C.3: Base de Datos de *AraWord*²

C.3.3. Diseño de la Aplicación

En la figura C.4 puede apreciarse la división en subsistemas de los diferentes módulos que componen la aplicación. El diseño de la aplicación se ha realizado con tecnologías Web utilizando HTML, CSS y JavaScript.

C. Descripción detallada del proceso de desarrollo

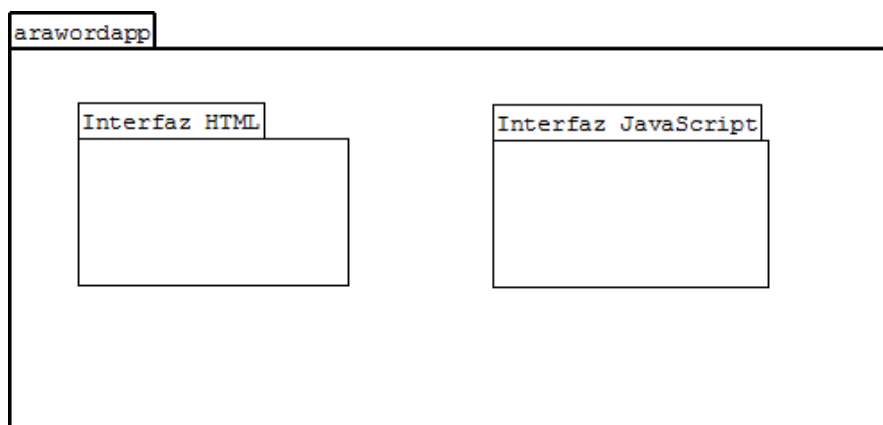


Figura C.4: Diagrama de paquetes de la Aplicación *AraWord*²

La aplicación se compone de una Interfaz HTML y de una Interfaz JavaScript:

- **Interfaz HTML:** Contiene todos los ficheros que conforman la aplicación. Estos forman la interfaz con la que el usuario interactúa. Consta de los siguientes ficheros:
 - Ficheros HTML: index.html, Login.html, myData.html, myPictograms.html, newUser.html, pinta_frase.html.
 - Hojas de estilo CSS: bootstrap.css, bootstrap-responsive.css, smallDevices.css
- **Interfaz JavaScript:** Contiene todos los métodos para realizar las invocaciones a la Interfaz de Servicios y el modelo de clases que se presenta en la Interfaz HTML.
 - webservices.functions.js: Funciones para realizar la invocación al API de Servicios.
 - utils2.js: Modelo de Objetos y funciones para gestionar la Interfaz HTML.

C.3.4. Servidor Web

El servidor Web utilizado para realizar el despliegue de la Interfaz de Servicios y de la aplicación *AraW²ord* ha sido Apache Tomcat en su versión 7.0.40.

Apache Tomcat no es un servidor de aplicaciones sino un contenedor de servlets desarrollado en Apache Software Foundation¹. Tomcat es un Servidor Web con soporte de servlets o JSPs.

Una vez instalado, podemos diferenciar la siguiente jerarquía de ficheros:

- **bin** - arranque, cierre, y otros scripts y ejecutables.

¹<http://www.apache.org/>

C. Descripción detallada del proceso de desarrollo

- **common** - clases comunes que pueden utilizar Catalina y las aplicaciones web.
- **conf** - ficheros XML y los correspondientes DTD para la configuración de Tomcat.
- **logs** - logs de Catalina y de las aplicaciones.
- **server** - clases utilizadas solamente por Catalina.
- **shared** - clases compartidas por todas las aplicaciones web.
- **webapps** - directorio que contiene las aplicaciones web.
- **work** - almacenamiento temporal de ficheros y directorios.

Configuración de Tomcat

Para la instalación de Tomcat se habilita el puerto 8090 en el servidor Alkaid del DIIS. Para realizar la configuración se modifica el fichero server.xml contenido en la carpeta conf y se confira el puerto del servidor, el puerto del conector y el conector AJP².

Para arrancar y parar el servidor se proporcionar dos scripts, startup.sh y shutdown.sh, contenidos en la carpeta bin.

Despliegue en Tomcat

Para realizar el despliegue de los servicios Web en Apache Tomcat ser crea un fichero WAR³ que contiene todos los métodos disponibles y se coloca en la carpeta webapps.

C.4. Herramientas Utilizadas

Durante la realización de este proyecto se han utilizado diferentes herramientas y tecnologías para el desarrollo del software y tareas de documentación. A continuación listamos dichas herramientas y su propósito.

- **Sistemas operativos**
 - Windows 7
 - Ubuntu GNU/Linux 12.04
 - Android 4.0.4
- **Propósito general**
 - Google Chrome 29.0.1547.57 m
- **Análisis y diseño**
 - Dia Diagrams 0.97.2: Generador de diagramas

²Conector que se comunica con un conector Web usando el protocolo AJP

³Web Application Archive

C. Descripción detallada del proceso de desarrollo

- Google Calendar: Planificación de reuniones
- Drive Document: Elaboración de actas

■ Desarrollo y pruebas

- Eclipse 3.7: Entorno de desarrollo de los servicios web y conexión con base de datos.
- Apache Axis 2: Motor y contenedor de servicios web.
- SoapUI 4.5.1: Aplicación de licencia libre para testear servicios web para arquitecturas orientadas a servicios (SOA).
- Java SDK 1.6.0_33: Máquina virtual Java
- Notepad++ 6.1.5: Entorno de desarrollo para la vista (HTML+Javascript)
- WinSCP 5.1.5: cliente SFTP gráfico para Windows que emplea SSH.
- PuTTY: es un cliente SSH, Telnet, rlogin, y TCP raw con licencia libre
- MySQL Query Browser 1.1.20: Servidor de bases de datos
- Apache Tomcat 7.0.4: servidor web con soporte de servlets y JSPs (usado como servidor de aplicaciones web)
- Servidor HTTP Apache: servidor web HTTP de código abierto (usado como contenedor de recursos para almacenar los pictogramas)
- OpenShift

■ Documentación

- TeXstudio 2.5.2 : Suite para compilar y editar con Latex (usado para la elaboración de la memoria y los anexos)
- Dia Diagrams 0.97.2: Generador de diagramas
- Microsoft Office Project: Herramienta para los diagramas de Gantt

C.5. Pruebas

El objetivo de las pruebas es verificar y validar el software, descubrir defectos que no han podido ser detectados con anterioridad y encontrar otros con poco esfuerzo y una alta probabilidad.

A lo largo del desarrollo del software se han ido haciendo tests y pruebas, las cuales podemos clasificarlas en: pruebas unitarias, pruebas de integración y pruebas de sistema y usuario.

C. Descripción detallada del proceso de desarrollo

Pruebas unitarias

Estas pruebas se han llevado a cabo después de la realización de cada módulo, ya que dichas pruebas permiten probar un módulo software de manera independiente. Las ventajas de este tipo de pruebas son que se suelen formar con un bloque de código muy básico y relativamente pequeño y con ellas se pueden detectar cálculos incorrectos, estructuras de datos incorrectas, controles de flujo incorrectos, etc.

Para comprobar el correcto funcionamiento de los servicios web se ha usado un software llamado SoapUI⁴ que te permite crear un proyecto con los servicios web publicados. Una vez definimos los proyectos a partir de los WSDL se definieron varios documentos XML de prueba para testar los servicios web.

La figura C.5 muestra una ejecución en soapUI. En verde se muestran las operaciones disponibles, en azul la petición al servicio Web y en rojo la respuesta del servicio Web.

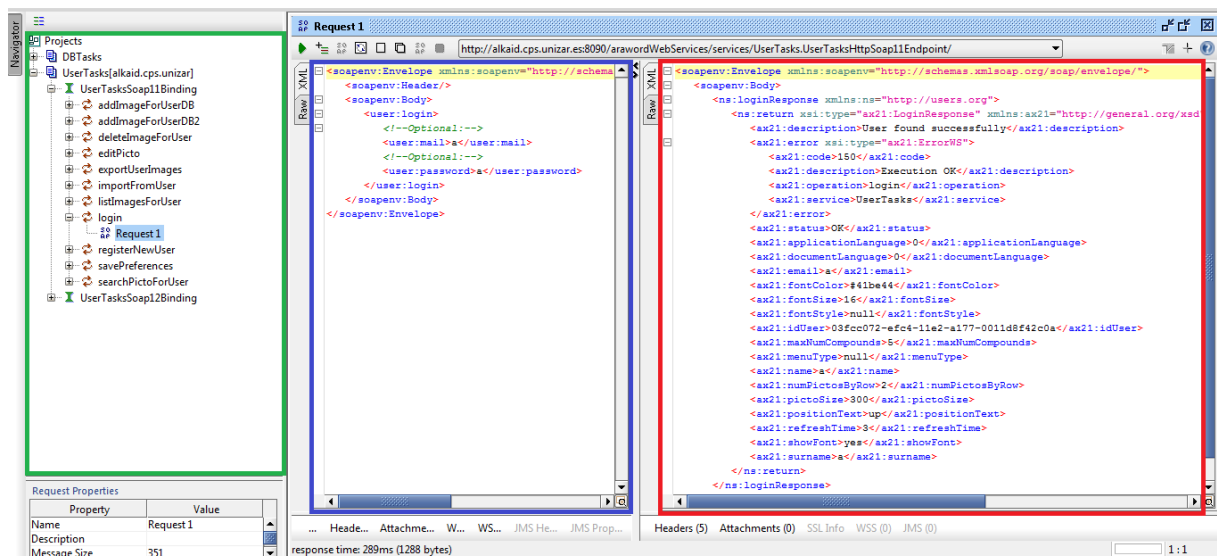


Figura C.5: Pruebas con SoapUI

Pruebas de integración

Una vez superadas las pruebas unitarias se han realizado pruebas de integración, las cuales verifican el correcto ensamblaje de los diferentes módulos. Estas pruebas aseguran que los componentes interactúan correctamente. Dichas pruebas se han llevado a cabo de manera incremental cada vez que se finalizaba un subsistema.

Para la realización de estas pruebas se han diseñado clientes HTML sencillos que testan cada uno de los servicios Web. En ellos se comprueba que la integración del servicio Web con el cliente HTML y con la Base de Datos funciona de forma correcta.

⁴<http://www.soapui.org/>

C. Descripción detallada del proceso de desarrollo

La figura C.6 muestra un ejemplo de un cliente HTML para testar el servicio Web de listado de imágenes propias de usuario.

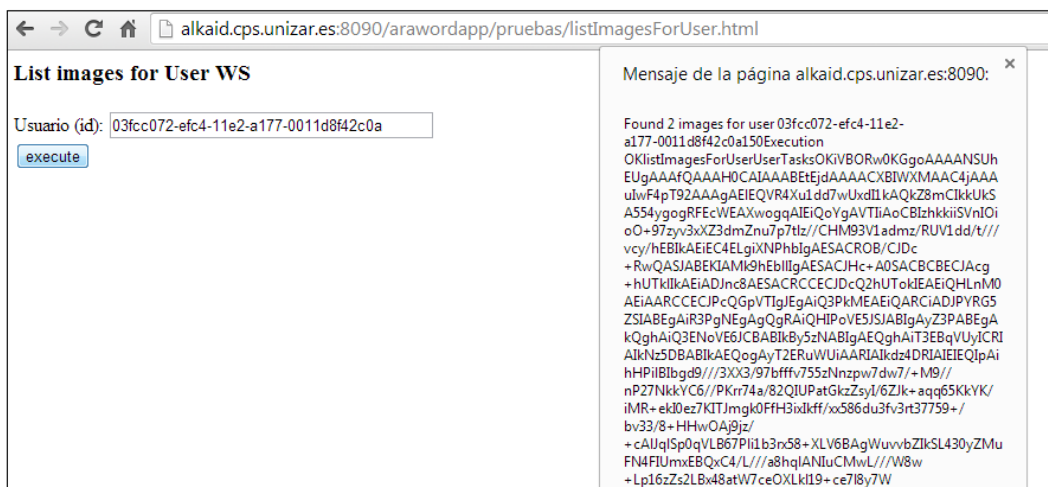


Figura C.6: Pruebas de integración

Pruebas de sistema y usuario

Estas pruebas tiene como finalidad comprobar la integración del sistema visto desde el punto de vista global y suelen estar basadas en las especificaciones técnicas. Una vez finalizada la aplicación se han realizado pruebas de posibles casos reales y también pruebas atípicas para estudiar el rendimiento.

Se han realizado diferentes pruebas con el comportamiento de la detección de las palabras compuestas ya que ello implica realizar bastantes llamadas a los servicios web y esto es bastante costoso. Se ha observado que con palabras compuestas de más de 3 palabras el servicio web tarda bastante en darnos el pictograma asociado y puede ser molesto para el usuario.

Entre las pruebas de carga podemos destacar la carga de una imagen muy grande, ya que el sistema tiene que redimensionar la imagen para guardarla en Base de Datos. Se observa que con imágenes mayores de 5MBytes el tiempo que tarde es bastante alto.

Otras pruebas de carga y de usuario han consistido en la comprobación del funcionamiento del software cuando varios usuarios ejecutan la aplicación a la vez.

Anexo D.

Servicios Web Desarrollados

El presente documento tiene la finalidad de describir los servicios Web publicados.

login

La función login permite al usuario realizar la autenticación en la aplicación y genera una nueva sesión. A continuación se detallan la estructura y los campos de los mensajes de petición y respuesta.

Servicio	WsdL: http://alkaid.cps.unizar.es:8090/arawordWebServices/services/UserTasks?wsdl		
Operación	login		
Descripción	La operación realiza la autenticación en la aplicación		
Tipo/Versión	SOAP v1.1 y v1.2		
Devuelve	loginResponse		
Parámetro	Tipo	Input/Output	Descripción
mail	XML.STRING	Input/-	Correo electrónico del usuario que lo identifica de forma unívoca
password	XML.STRING	Input/-	Password asociado al mail

Ejemplo de petición XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:user="http://users.org">
  <soapenv:Header/>
  <soapenv:Body>
    <user:login>
      <user:mail>a</user:mail>
      <user:password>a</user:password>
    </user:login>
  </soapenv:Body>
</soapenv:Envelope>
```

D. Servicios Web Desarrollados

loginResponse

Parámetro	Tipo	Input/Output	Descripción
description	XML_STRING	-/Output	Descripción de cómo ha ido la operación
status	XML_STRING	-/Output	Ver Códigos de error
idUser	XML_STRING	-/Output	id del usuario
error	error	-/Output	Nodo de error

error	Nodo de error		
Parámetro	Tipo	Input/Output	Descripción
code	XML_INT	-/Output	Ver Códigos de error
description	XML_STRING	-/Output	Descripción del error
service	XML_STRING	-/Output	Servicio que provoca el error
operation	XML_STRING	-/Output	Operación que provoca el error

Ejemplo de respuesta XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:loginResponse xmlns:ns="http://users.org">
      <ns:return xsi:type="ax21:LoginResponse" xmlns:ax21="http://general.org/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ax21:description>User found successfully</ax21:description>
        <ax21:error xsi:type="ax21:ErrorWS">
          <ax21:code>150</ax21:code>
          <ax21:description>Execution OK</ax21:description>
          <ax21:operation>login</ax21:operation>
          <ax21:service>UserTasks</ax21:service>
        </ax21:error>
        <ax21:status>OK</ax21:status>
        <ax21:applicationLanguage>0</ax21:applicationLanguage>
        <ax21:documentLanguage>0</ax21:documentLanguage>
        <ax21:email>a</ax21:email>
        <ax21:fontColor>#ff0000</ax21:fontColor>
        <ax21:fontSize>16</ax21:fontSize>
        <ax21:fontStyle>Calibri</ax21:fontStyle>
        <ax21:idUser>03fcc072-efc4-11e2-a177-0011d8f42c0a</ax21:idUser>
        <ax21:maxNumCompounds>5</ax21:maxNumCompounds>
        <ax21:menuType>text</ax21:menuType>
      </ns:return>
    </ns:loginResponse>
  </soapenv:Body>
</soapenv:Envelope>
```


D. Servicios Web Desarrollados

```
<ax21:name>a</ax21:name>
<ax21:numPictosByRow>3</ax21:numPictosByRow>
<ax21:pictoSize>100</ax21:pictoSize>
<ax21:positionText>up</ax21:positionText>
<ax21:refreshTime>1</ax21:refreshTime>
<ax21:showFont>yes</ax21:showFont>
<ax21:surname>a</ax21:surname>
</ns:return>
</ns:loginResponse>
</soapenv:Body>
</soapenv:Envelope>
```

registerNewUser

La función registerNewUser permite al usuario registrarse en la aplicación de forma que en los sucesivos accesos puede autenticarse y recuperar su perfil. A continuación se detallan la estructura y los campos de los mensajes de petición y respuesta de la función registerNewUser.

Servicio	Wsdl: http://alkaid.cps.unizar.es:8090/arawordWebServices/services/UserTasks?wsdl		
Operación	registerNewUser		
Descripción	La operación registerNewUser registra un nuevo usuario en la base de datos que se identifica de forma unívoca por el mail		
Tipo/Versión	SOAP v1.1 y v1.2		
Devuelve	registerNewUserResponse		
Parámetro	Tipo	Input/Output	Descripción
name	XML.STRING	Input/-	Nombre del usuario
surname	XML.STRING	Input/-	Apellidos del usuario
mail	XML.STRING	Input/-	Correo electrónico del usuario que lo identifica de forma unívoca
password	XML.STRING	Input/-	Password asociado al mail

Ejemplo de petición XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:user="http://users.org">
  <soapenv:Header/>
  <soapenv:Body>
    <user:registerNewUser>
      <user:name> Aaron </user:name>
      <user:surname> Torrijos </user:surname>
      <user:mail> aaron.torri@gmail.com </user:mail>
      <user:password> passwordAaron </user:password>
    </user:registerNewUser>
  </soapenv:Body>
</soapenv:Envelope>
```

D. Servicios Web Desarrollados

registerNewUserResponse

Parámetro	Tipo	Input/Output	Descripción
description	XML_STRING	-/Output	Descripción de cómo ha ido la operación
status	XML_STRING	-/Output	Ver Códigos de error
error	error	-/Output	Nodo de error

error	Nodo de error		
Parámetro	Tipo	Input/Output	Descripción
code	XML_INT	-/Output	Ver Códigos de error
description	XML_STRING	-/Output	Descripción del error
service	XML_STRING	-/Output	Servicio que provoca el error
operation	XML_STRING	-/Output	Operación que provoca el error

Ejemplo de respuesta XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:registerNewUserResponse xmlns:ns="http://users.org">
      <ns:return xsi:type="ax21:RegisterNewUserResponse" xmlns:ax21="http://general.org/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ax21:description>User added successfully</ax21:description>
        <ax21:error xsi:type="ax21:ErrorWS">
          <ax21:code>100</ax21:code>
          <ax21:description>Database Insert OK</ax21:description>
          <ax21:operation>registerNewUser</ax21:operation>
          <ax21:service>UserTasks</ax21:service>
        </ax21:error>
        <ax21:status>OK</ax21:status>
      </ns:return>
    </ns:registerNewUserResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

D. Servicios Web Desarrollados

addImageForUserDB2

La función addImageForUserDB2 permite al usuario añadir una imagen a su lista de imágenes personales. A continuación se detallan la estructura y los campos de los mensajes de petición y respuesta de la función addImageForUserDB2.

Servicio	Wsdl: http://alkaid.cps.unizar.es:8090/arawordWebServices/services/UserTasks?wsdl		
Operación	addImageForUserDB2		
Descripción	La operación addImageForUserDB2 inserta una imagen seleccionada por el usuario a su colección		
Tipo/Versión	SOAP v1.1 y v1.2		
Devuelve	addImageForUserDBResponse		
Parámetro	Tipo	Input/Output	Descripción
idImage	XML.STRING	Input/-	Identificador de la imagen a añadir
base64	XML.STRING	Input/-	Codificación de la imagen en base 64
language	XML.STRING	Input/-	Lenguaje de la palabra
type	XML.STRING	Input/-	Tipo de palabra
word	XML.STRING	Input/-	Palabra asociada a la imagen
idUser	XML.STRING	Input/-	Identificador del usuario que inserta la imagen

Ejemplo de petición XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:user="http://users.org">
  <soapenv:Header/>
  <soapenv:Body>
    <user:addImageForUserDB2>
      <user:idImage>02548.png</user:idImage>
      <user:base64>iVBORwOKGgoAAAANSU...</user:base64>
      <user:language>Castellano</user:language>
      <user:type>nombreComun</user:type>
      <user:word>misDatos</user:word>
      <user:idUser>71032b9b-882b-11e2-9031-b5f4a97139cc</user:idUser>
    </user:addImageForUserDB2>
  </soapenv:Body>
</soapenv:Envelope>
```

addImageForUserDBResponse

Parámetro	Tipo	Input/Output	Descripción
description	XML.STRING	-/Output	Descripción de cómo ha ido la operación
status	XML.STRING	-/Output	Ver Códigos de error
error	error	-/Output	Nodo de error

D. Servicios Web Desarrollados

Ejemplo de respuesta XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:addImageForUserDBResponse xmlns:ns="http://users.org">
      <ns:return xsi:type="ax21:RegisterNewUserResponse" xmlns:ax21="http://general.org/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ax21:description>Image added successfully</ax21:description>
      >
      <ax21:error xsi:type="ax21:ErrorWS">
        <ax21:code>100</ax21:code>
        <ax21:description>Database Insert OK</ax21:description>
        <ax21:operation> addImageForUserDB</ax21:operation>
        <ax21:service>UserTasks</ax21:service>
      </ax21:error>
      <ax21:status>OK</ax21:status>
    </ns:return>
  </ns:addImageForUserDBResponse>
</soapenv:Body>
</soapenv:Envelope>
```

deleteImageForUser

La función deleteImageForUser permite al usuario borrar una imagen de su lista de imágenes personales. A continuación se detallan la estructura y los campos de los mensajes de petición y respuesta de la función deleteImageForUser.

Servicio	WsdL: http://alkaid.cps.unizar.es:8090/arawordWebServices/services/UserTasks?wsdl		
Operación	deleteImageForUser		
Descripción	La operación deleteImageForUser borra una imagen de la colección del usuario		
Tipo/versión	SOAP v1.1 y v1.2		
Devuelve	deleteImageForUserResponse		
Parámetro	Tipo	Input/Output	Descripción
name	XML.STRING	Input/-	Nombre de la imagen a borrar
idUser	XML.STRING	Input/-	id del usuario que borra la imagen

Ejemplo de petición XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:user="http://users.org">
  <soapenv:Header/>
  <soapenv:Body>
    <user:deleteImageForUser>
      <user:name>descarga.png</user:name>
      <user:idUser>03fcc072-efc4-11e2-a177-0011d8f42c0a</user:idUser>
    </user:deleteImageForUser>
  </soapenv:Body>
</soapenv:Envelope>
```

D. Servicios Web Desarrollados

deleteImageForUserResponse

Parámetro	Tipo	Input/Output	Descripción
description	XML_STRING	-/Output	Descripción de cómo ha ido la operación
status	XML_STRING	-/Output	Ver Códigos de error
error	error	-/Output	Nodo de error

Ejemplo de respuesta XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:deleteImageForUserResponse xmlns:ns="http://users.org">
      <ns:return xsi:type="ax21:DeleteImageForUserResponse" xmlns:ax21="http://general.org/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ax21:description>Image deleted</ax21:description>
        <ax21:error xsi:type="ax21:ErrorWS">
          <ax21:code>101</ax21:code>
          <ax21:description>Database Delete</ax21:description>
          <ax21:operation>deleteImageForUser</ax21:operation>
          <ax21:service>UserTasks</ax21:service>
        </ax21:error>
        <ax21:status>OK</ax21:status>
      </ns:return>
    </ns:deleteImageForUserResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

listImagesForUser

La función listImagesForUser permite al usuario ver el listado de las imágenes de su colección. A continuación se detallan la estructura y los campos de los mensajes de petición y respuesta de la función listImagesForUser.

Servicio	Wsdl: http://alkaid.cps.unizar.es:8090/arawordWebServices/services/UserTasks?wsdl		
Operación	listImagesForUser		
Descripción	La operación listImagesForUser muestra una lista de todas las imágenes de un usuario		
Tipo/Versión	SOAP v1.1 y v1.2		
Devuelve	listImagesForUserResponse		
Parámetro	Tipo	Input/Output	Descripción
idUser	XML_STRING	Input/-	identificador del usuario

D. Servicios Web Desarrollados

Ejemplo de petición XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:user="http://users.org">
  <soapenv:Header/>
  <soapenv:Body>
    <user:listImagesForUser>
      <user:idUser>03fcc072-efc4-11e2-a177-0011d8f42c0a</user:idUser>
    </user:listImagesForUser>
  </soapenv:Body>
</soapenv:Envelope>
```

listImagesForUserResponse

Parámetro	Tipo	Input/Output	Descripción
ImagesArray	ImagesArray	-/Output	Lista que contiene las imágenes propias de un usuario
description	XML.STRING	-/Output	Descripción de cómo ha ido la operación
status	XML.STRING	-/Output	ver Códigos de error
error	error	-/Output	Nodo de error

Parámetro	Tipo	Input/Output	Descripción
base64	XML.STRING	-/Output	Codificación de la imagen en base64
language	XML.STRING	-/Output	Lenguaje de la palabra asociada a la imagen
name	XML.STRING	-/Output	Nombre de la imagen
type	XML.STRING	-/Output	Tipo de palabra de la imagen
word	XML.STRING	-/Output	Nombre de la realidad que representa la imagen

Ejemplo de respuesta XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:listImagesForUserResponse xmlns:ns="http://users.org">
      <ns:return xsi:type="ax21:ListImagesForUserResponse" xmlns:ax21="http://general.org/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ax21:description>Found 2 images for user 03fcc072-efc4-11e2-a177-0011d8f42c0a</ax21:description>
        <ax21:error xsi:type="ax21:ErrorWS">
          <ax21:code>150</ax21:code>
          <ax21:description>Execution OK</ax21:description>
          <ax21:operation>listImagesForUser</ax21:operation>
        </ax21:error>
      </ns:return>
    </ns:listImagesForUserResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

D. Servicios Web Desarrollados

```

        <ax21:service>UserTasks</ax21:service>
    </ax21:error>
    <ax21:status>OK</ax21:status>
    <ax21:imagesArray>
        <ax21:base64>iVBORwOKGgoAAAANSUhEUgAAAfQAAAHOCAI...</ax21:base64>
        <ax21:language>0</ax21:language>
        <ax21:name>misDatos.png</ax21:name>
        <ax21:type>0</ax21:type>
        <ax21:word>aaron</ax21:word>
    </ax21:imagesArray>
    <ax21:imagesArray>
        <ax21:base64>/9j/4AAQSkZJRgABAQEASABIAAD/4QD8RXh...</ax21:base64>
        <ax21:language>6</ax21:language>
        <ax21:name>ok.png</ax21:name>
        <ax21:type>5</ax21:type>
        <ax21:word>ok</ax21:word>
    </ax21:imagesArray>
    </ns:return>
</ns:listImagesForUserResponse>
</soapenv:Body>
</soapenv:Envelope>

```

searchPictoForUser

La función searchPictoForUser permite al usuario buscar una imagen seleccionada por el usuario de su colección y del servidor de pictos. A continuación se detallan la estructura y los campos de los mensajes de petición y respuesta de la función searchPictoForUser.

Servicio	WsdL: http://alkaid.cps.unizar.es:8090/arawordWebServices/services/UserTasks?wsdl		
Operación	searchPictoForUser		
Descripción	La operación searchPictoForUser busca una imagen seleccionada por el usuario de su colección y del servidor de pictogramas		
Tipo/Versión	SOAP v1.1 y v1.2		
Devuelve	searchPictoForUserResponse		
Parámetro	Tipo	Input/Output	Descripción
word	XML.STRING	Input/-	Nombre de la imagen a buscar
idUser	XML.STRING	Input/-	identificador del usuario (opcional)
language	XML.STRING	Input/-	Lenguaje de la palabra a buscar

D. Servicios Web Desarrollados

Ejemplo de petición XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
  envelope/" xmlns:user="http://users.org">
  <soapenv:Header/>
  <soapenv:Body>
    <user:searchPictoForUser>
      <user:word>abeja</user:word>
      <user:idUser></user:idUser>
      <user:language>0</user:language>
    </user:searchPictoForUser>
  </soapenv:Body>
</soapenv:Envelope>
```

searchPictoForUserForUserResponse

Parámetro	Tipo	Input/Output	Descripción
pictos	pictos	-/Output	Lista que contiene las imágenes encontradas
description	XML_STRING	-/Output	Descripción de cómo ha ido la operación
status	XML_STRING	-/Output	ver Códigos de error
error	error	-/Output	Nodo de error

Parámetro	Tipo	Input/Output	Descripción
pictos	Lista que contiene las imágenes encontradas		
url	XML_STRING	-/Output	url de la imagen. En caso que sea una imagen propia se muestra la codificación en base64
type	XML_STRING	-/Output	Tipo de palabra de la imagen
wordType	XML_STRING	-/Output	[WEB,USER] dependiendo de si la imagen es propia del usuario o no

Ejemplo de respuesta XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
  envelope/">
  <soapenv:Body>
    <ns:searchPictoForUserResponse xmlns:ns="http://users.org">
      <ns:return xsi:type="ax21:SearchPictoForUserResponse" xmlns:
        ax21="http://general.org/xsd" xmlns:xsi="http://www.w3.org
        /2001/XMLSchema-instance">
        <ax21:description>Found 2 images for user</ax21:description>
        <ax21:error xsi:type="ax21:ErrorWS">
          <ax21:code>150</ax21:code>
          <ax21:description>Execution OK</ax21:description>
          <ax21:operation>searchPictoForUser</ax21:operation>
          <ax21:service>UserTasks</ax21:service>
        </ax21:error>
      </ns:return>
    </ns:searchPictoForUserResponse>
  </soapenv:Body>
</soapenv:Envelope>
```


D. Servicios Web Desarrollados

```
<ax21:status>OK</ax21:status>
<ax21:pictos xsi:type="ax21:PictoArrayView">
  <ax21:pvList>
    <ax21:type>WEB</ax21:type>
    <ax21:url>http://www.gidhe.es/pictostorrijos/2/2239.
      png</ax21:url>
    <ax21:wordType>0</ax21:wordType>
  </ax21:pvList>
  <ax21:pvList>
    <ax21:type>WEB</ax21:type>
    <ax21:url>http://www.gidhe.es/pictostorrijos/2/24823.png
      </ax21:url>
    <ax21:wordType>0</ax21:wordType>
  </ax21:pvList>
</ax21:pictos>
<ax21:possibleCompound>>false</ax21:possibleCompound>
<ax21:possibleCompoundFromBegin>>false</ax21:
  possibleCompoundFromBegin>
<ax21:possibleCompoundFromEnd>>false</ax21:
  possibleCompoundFromEnd>
<ax21:possibleCompoundFromMiddle>>false</ax21:
  possibleCompoundFromMiddle>
<ax21:type>0</ax21:type>
</ns:return>
</ns:searchPictoForUserResponse>
</soapenv:Body>
</soapenv:Envelope>
```

D. Servicios Web Desarrollados

savePreferences

La operación savePreferences permite al usuario modificar los siguientes parámetros: nº de pictogramas por línea, tamaño de la fuente, posición del texto, color del texto, mostrar el texto, nº de compuestas, tamaño de imágenes, estilo de la fuente, lenguaje del documento, lenguaje de la aplicación, tipo de menu, tiempo de refresco, mostrar borde. A continuación se detallan la estructura y los campos de los mensajes de petición y respuesta de la función savePreferences.

Servicio	Wsdl: http://alkaid.cps.unizar.es:8090/arawordWebServices/services/UserTasks?wsdl		
Operación	savePreferences		
Descripción	La operación savePreferences permite al usuario modificar las preferencias		
Tipo/Versión	SOAP v1.1 y v1.2		
Devuelve	savePreferencesResponse		
Parámetro	Tipo	Input/Output	Descripción
idUser	XML_STRING	Input/-	identificador del usuario
numPictosByRow	XML_STRING	Input/-	Número de pictogramas a mostrar por línea
fontSize	XML_STRING	Input/-	Tamaño de la fuente
textPosition	XML_STRING	Input/-	Posición del texto (encima o debajo de la imagen)
fontColor	XML_STRING	Input/-	Color de la fuente
showText	XML_STRING	Input/-	Mostrar el texto o no
maxNumCompounds	XML_STRING	Input/-	Máximo número de palabras que forman una compuesta
pictoSize	XML_STRING	Input/-	Tamaño del pictograma
fontStyle	XML_STRING	Input/-	Estilo de la fuente
documentLanguage	XML_STRING	Input/-	Idioma del documento
applicationLanguage	XML_STRING	Input/-	Idioma de la aplicación
menuType	XML_STRING	Input/-	Tipo de menu (texto o pictogramas)
refreshTime	XML_STRING	Input/-	Tiempo de refresco de los pictogramas
borderColor	XML_STRING	Input/-	Mostrar si o no el borde del pictograma

Ejemplo de petición XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:user="http://users.org">
  <soapenv:Header/>
  <soapenv:Body>
    <user:savePreferences>
      <user:idUser>03fcc072-efc4-11e2-a177-0011d8f42c0a</user:idUser>
      <user:numPictosRow>4</user:numPictosRow>
      <user:fontSize>16</user:fontSize>
      <user:textPosition>up</user:textPosition>
      <user:fontColor>#ff0000</user:fontColor>
      <user:showText>yes</user:showText>
      <user:maxNumCompounds>3</user:maxNumCompounds>
      <user:pictoSize>100</user:pictoSize>
      <user:fontStyle>Calibri</user:fontStyle>
      <user:documentLanguage>0</user:documentLanguage>
      <user:applicationLanguage>0</user:applicationLanguage>
      <user:menuType>picto</user:menuType>
      <user:refreshTime>3</user:refreshTime>
    </user:savePreferences>
  </soapenv:Body>
</soapenv:Envelope>
```

D. Servicios Web Desarrollados

savePreferencesResponse

Parámetro	Tipo	Input/Output	Descripción
description	XML_STRING	-/Output	Descripción de cómo ha ido la operación
status	XML_STRING	-/Output	Ver Códigos de error
error	error	-/Output	Nodo de error

Ejemplo de respuesta XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:savePreferencesResponse xmlns:ns="http://users.org">
      <ns:return xsi:type="ax21:SavePreferencesResponse" xmlns:ax21="http://general.org/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ax21:description>Preferences changed successfully for user 03fcc072-efc4-11e2-a177-0011d8f42c0a</ax21:description>
        <ax21:error xsi:type="ax21:ErrorWS">
          <ax21:code>150</ax21:code>
          <ax21:description>Execution OK</ax21:description>
          <ax21:operation>savePreferences</ax21:operation>
          <ax21:service>UserTasks</ax21:service>
        </ax21:error>
        <ax21:status>OK</ax21:status>
      </ns:return>
    </ns:savePreferencesResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

editPicto

La función editPicto permite al usuario cambiar el nombre de las imágenes pertenecientes a su colección. A continuación se detallan la estructura y los campos de los mensajes de petición y respuesta de la función editPicto.

Servicio	Wsdl: http://alkaid.cps.unizar.es:8090/arawordWebServices/services/UserTasks?wsdl		
Operación	editPicto		
Descripción	La operación editPicto permite al usuario cambiar el nombre de sus imágenes		
Tipo/Versión	SOAP v1.1 y v1.2		
Devuelve	editPictoResponse		
Parámetro	Tipo	Input/Output	Descripción
idUser	XML_STRING	Input/-	identificador del usuario
idName	XML_STRING	Input/-	identificador de la imagen
newWord	XML_STRING	Input/-	Nuevo nombre del pictograma

D. Servicios Web Desarrollados

Ejemplo de petición XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:user="http://users.org">
  <soapenv:Header/>
  <soapenv:Body>
    <user:editPicto>
      <user:newWord>datos</user:newWord>
      <user:idName>misDatos.png</user:idName>
      <user:idUser>03fcc072-efc4-11e2-a177-0011d8f42c0a</user:idUser>
    </user:editPicto>
  </soapenv:Body>
</soapenv:Envelope>
```

editPictoResponse

Parámetro	Tipo	Input/Output	Descripción
description	XML_STRING	-/Output	Descripción de cómo ha ido la operación
status	XML_STRING	-/Output	Ver Códigos de error
error	error	-/Output	Nodo de error

Ejemplo de respuesta XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:editPictoResponse xmlns:ns="http://users.org">
      <ns:return xsi:type="ax21:EditResponse" xmlns:ax21="http://general.org/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ax21:description>Picto edited successfully for user 03fcc072-efc4-11e2-a177-0011d8f42c0a</ax21:description>
        <ax21:error xsi:type="ax21:ErrorWS">
          <ax21:code>150</ax21:code>
          <ax21:description>Execution OK</ax21:description>
          <ax21:operation>editPicto</ax21:operation>
          <ax21:service>UserTasks</ax21:service>
        </ax21:error>
        <ax21:status>OK</ax21:status>
      </ns:return>
    </ns:editPictoResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

D. Servicios Web Desarrollados

Códigos de Error

Status	Descripción
OK	La operación ha finalizado correctamente
FAIL	Error en la operación
ERROR	Error grave

100-199	Códigos OK
100	Inserción en BD OK
101	Borrado en BD OK
150	Ejecución OK
200-299	Códigos ERROR
200	Error insertando en BD
201	Error eliminando en BD
250	Ejecución ERROR (Unexpected error)
251	Error al cerrar la conexión con BD
300-399	Códigos FAIL
300	Datos replicados en BD
301	Datos no presentes en BD
302	Datos no encontrados en BD

D. Servicios Web Desarrollados

Anexo E.

Manual de usuario

En este anexo se presenta el manual de usuario de la aplicación *AraW²ord*. Dado que se trata de una aplicación Web y no se requiere la instalación de ningún software adicional, simplemente se describen de forma detallada las operaciones que se pueden realizar. También se proporciona un storyboard¹ que proporciona una visión global de las tareas que se pueden ejecutar.

E.1. Manual de usuario

E.1.1. Página principal de la aplicación

La figura E.1 muestra la página principal de la aplicación *AraW²ord*, la cual está accesible mediante la url <http://alkaid.cps.unizar.es:8090/arawordapp/>. En esta pantalla se permite la edición de texto y se da acceso para entrar en la aplicación si se posee una cuenta de usuario o de crear una cuenta nueva. En todas las opciones se muestra el logo de AraWord que proporciona un enlace con la página principal, es decir, la página de edición de texto.

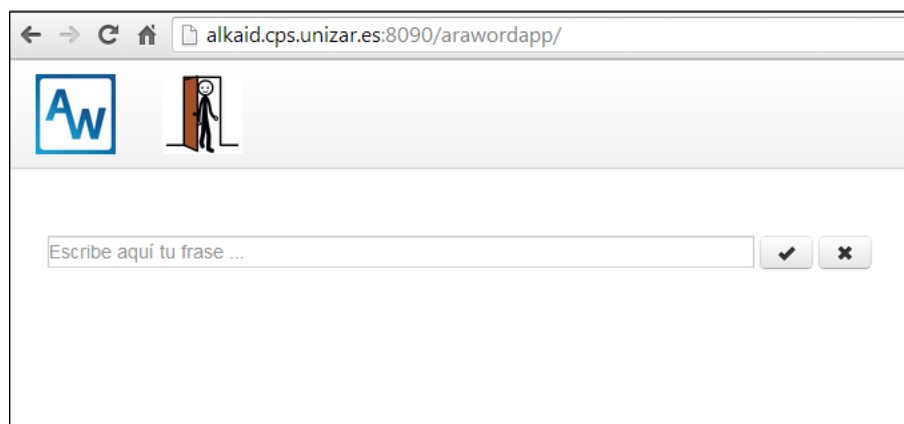


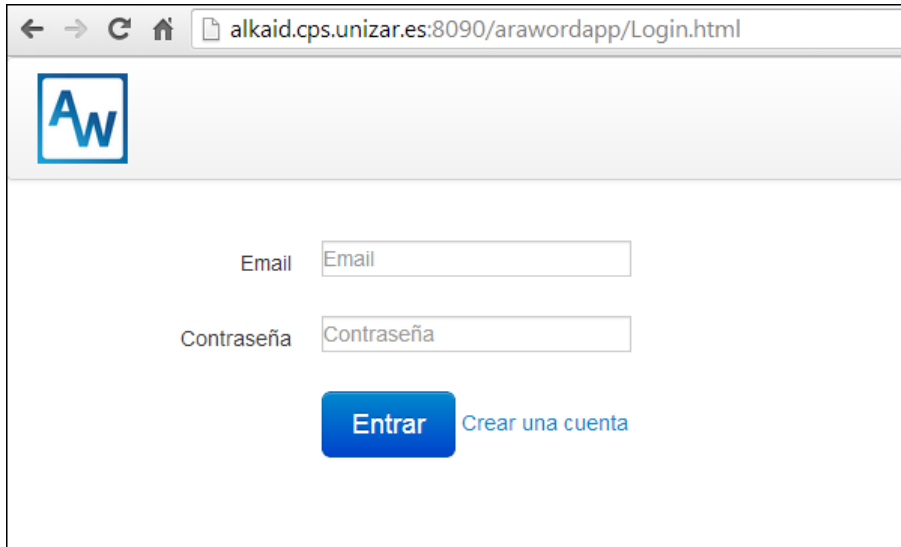
Figura E.1: Página de inicio de *AraW²ord*

¹Conjunto de ilustraciones mostradas en secuencia con el objetivo de servir de guía para entender un proceso

E. Manual de usuario

E.1.2. Registro o identificación de usuarios

La figura E.2 muestra la ventana de identificación de usuarios y proporciona un enlace para realizar el registro de nuevos usuarios. Para la identificación es necesario introducir el email y la contraseña correctamente.



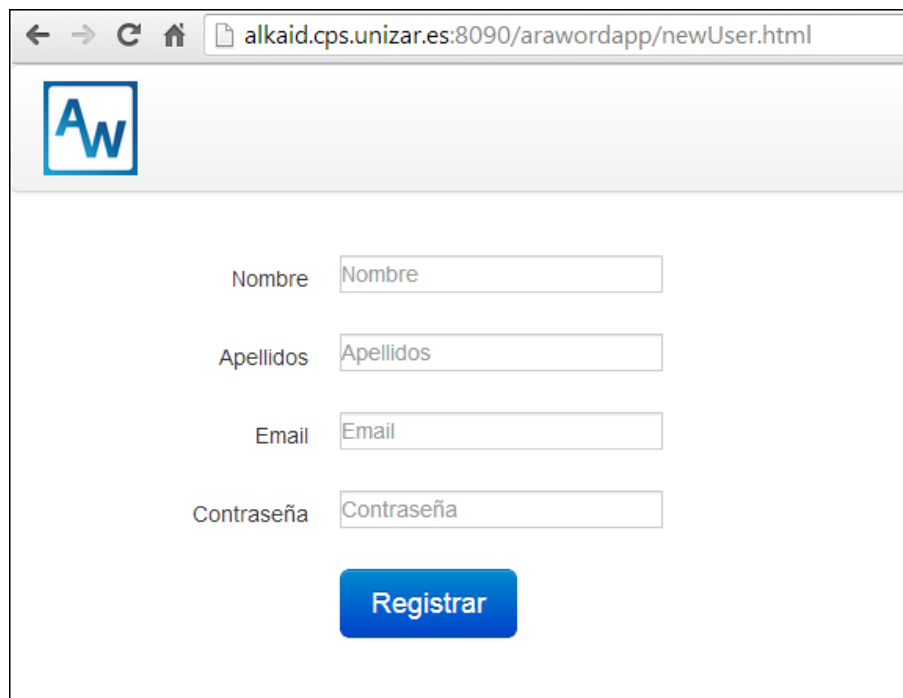
The image shows a web browser window with the address bar displaying "alkaid.cps.unizar.es:8090/arawordapp/Login.html". The page features a logo with the letters "Aw" in a blue square at the top left. Below the logo, there are two input fields: "Email" and "Contraseña". Underneath the "Contraseña" field, there is a blue button labeled "Entrar" and a blue text link labeled "Crear una cuenta".

Figura E.2: Ventana de login

E. Manual de usuario

E.1.3. Registro de nuevo usuario

La figura E.3 muestra la ventana para el registro de nuevos usuarios. Para el registro de un usuario nuevo se debe proporcionar nombre, apellidos, email y contraseña.

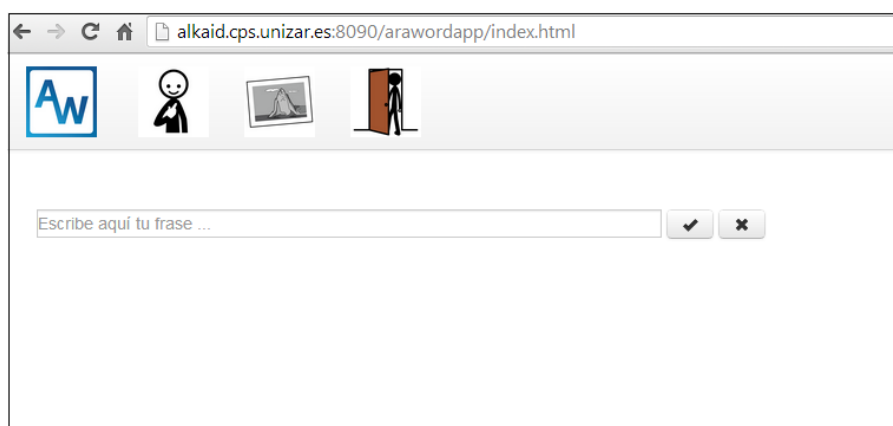


The screenshot shows a web browser window with the address bar containing 'alkaid.cps.unizar.es:8090/arawordapp/newUser.html'. The page features a header with a logo consisting of the letters 'Aw' in a blue square. Below the header, there is a registration form with four input fields: 'Nombre', 'Apellidos', 'Email', and 'Contraseña'. Each field is preceded by its respective label. At the bottom of the form, there is a blue button labeled 'Registrar'.

Figura E.3: Ventana de registro de usuarios

E.1.4. Opciones para usuarios registrados

Una vez que se accede a la aplicación como un usuario registrado aparecen unos menús adicionales. La figura E.4 muestra la pantalla de inicio de un usuario registrado. Los menús de izquierda a derecha son: pantalla de inicio, gestión de preferencias, gestión de pictogramas y cierre de sesión.



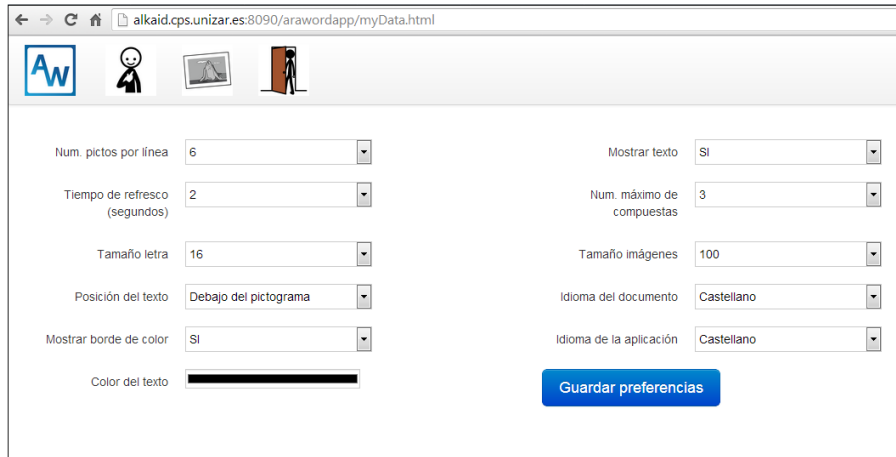
The screenshot shows a web browser window with the address bar containing 'alkaid.cps.unizar.es:8090/arawordapp/index.html'. The page features a header with the 'Aw' logo and three icons: a stick figure, a landscape with a mountain, and a person standing at a doorway. Below the header, there is a text input field with the placeholder text 'Escribe aquí tu frase ...'. To the right of the input field are two buttons: a checkmark icon and a close icon (an 'x' in a square).

Figura E.4: Pantalla de inicio para usuario registrados

E. Manual de usuario

Gestión de preferencias

La figura E.5 muestra la ventana de gestión de preferencias. En ella el usuario puede cambiar diferentes parámetros como el color del texto, el tamaño de las imágenes, etc.



The screenshot shows a web browser window with the address bar displaying "alkaid.cps.unizar.es:8090/arawordapp/myData.html". The page contains a settings form with the following fields:

Num. pictos por línea	6	Mostrar texto	SI
Tiempo de refresco (segundos)	2	Num. máximo de compuestas	3
Tamaño letra	16	Tamaño imágenes	100
Posición del texto	Debajo del pictograma	Idioma del documento	Castellano
Mostrar borde de color	SI	Idioma de la aplicación	Castellano
Color del texto	<input type="color" value="#000000"/>		

A blue button labeled "Guardar preferencias" is located at the bottom right of the form.

Figura E.5: Cambio de preferencias de un usuario

E. Manual de usuario

Gestión de pictogramas

La figura E.6 muestra la ventana de gestión de pictogramas propios. En esta ventana el usuario puede cargar nuevos pictogramas, modificar el nombre de pictogramas que tenga en su lista o eliminarlos.



Figura E.6: Gestión de pictogramas

E.1.5. Edición de texto

En la ventana principal se lleva a cabo el proceso de edición de texto. El usuario introduce texto en el cuadro de texto y los pictogramas aparecen de forma dinámica.

Cambio de nombre de un pictograma

La figura E.7 muestra el cambio de texto de un pictograma. Para realizar el cambio se debe pulsar el nombre del pictograma y aparece un pop-up con un cuadro de texto para introducir el nuevo nombre.

E. Manual de usuario

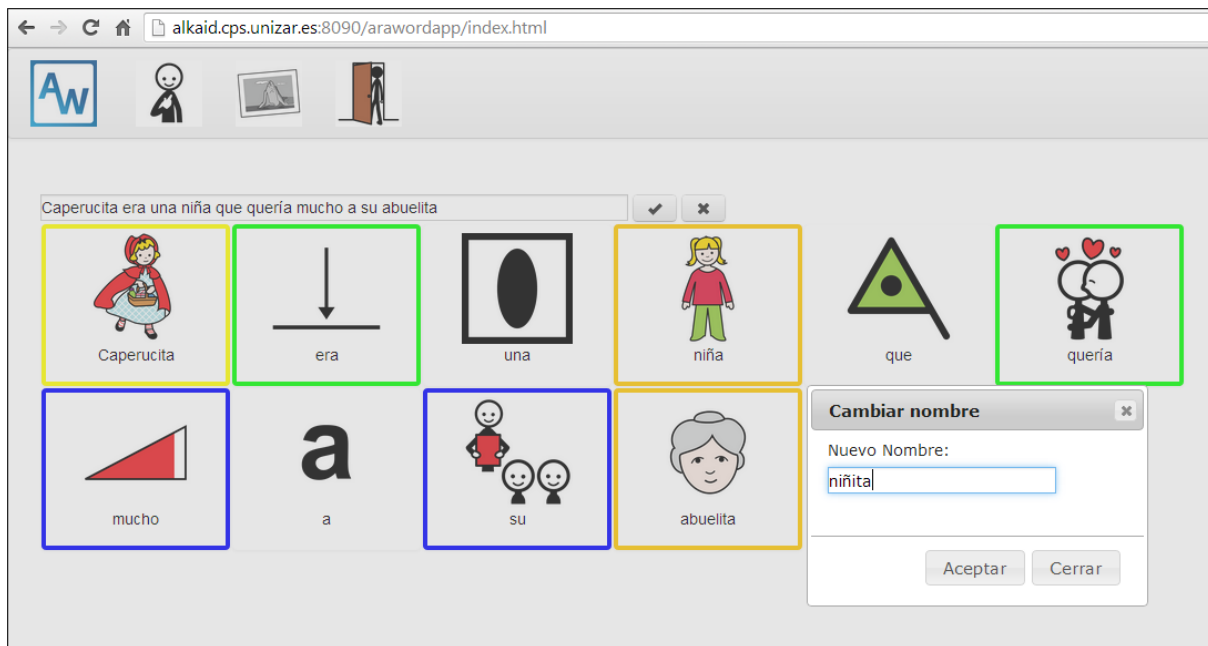


Figura E.7: Cambio de nombre

En la figura E.7 se cambia el texto 'niña' por 'niñita' y en la figura E.8 se muestra el resultado de este cambio.

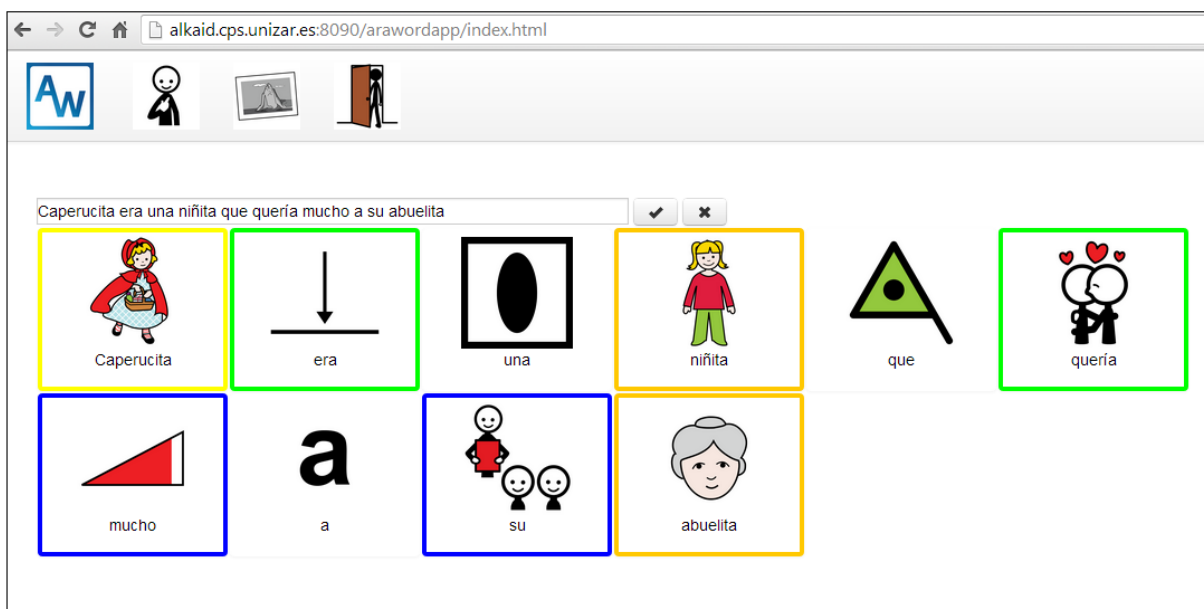


Figura E.8: Cambio de nombre realizado

E. Manual de usuario

Cambio de imagen

La figura E.9 muestra el cambio de imagen de un pictograma. Para ver las posibles imágenes asociadas a un concepto simplemente hay que pulsar en la imagen para pasar a la siguiente.

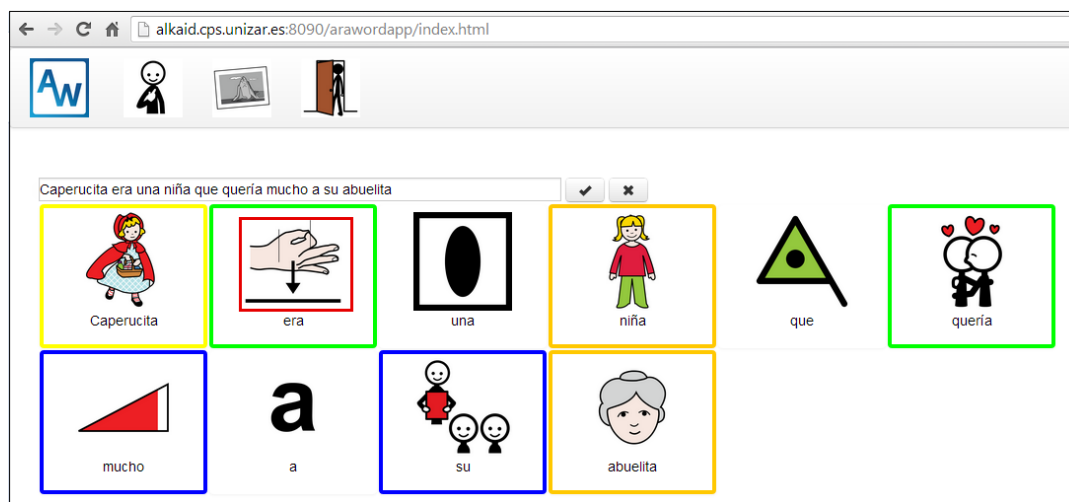
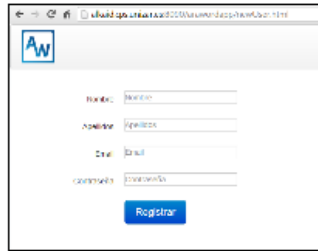


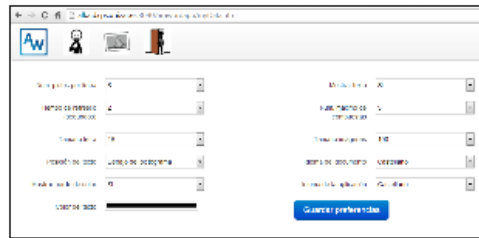
Figura E.9: Cambio de imagen

E.2. StoryBoard

La siguiente imagen muestra un storyboard de los procesos descritos anteriormente. Se muestran las diferentes pantallas y eventos para realizar cada una de las tareas.



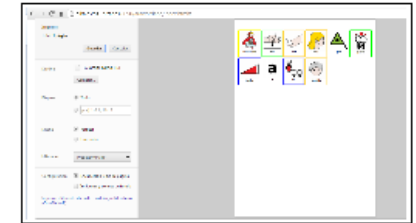
Pantalla de registro de nuevo usuario



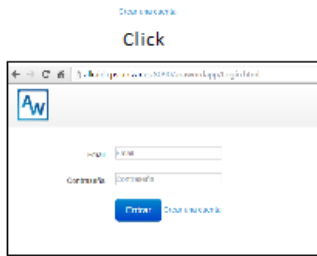
Pantalla de gestión de preferencias de usuario



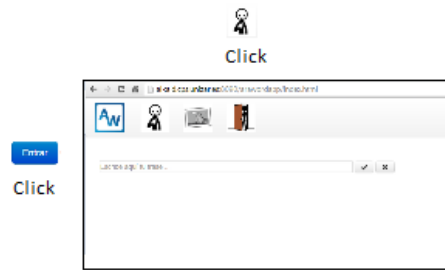
Cambio de texto de un pictograma



Opción de impresión de la frase



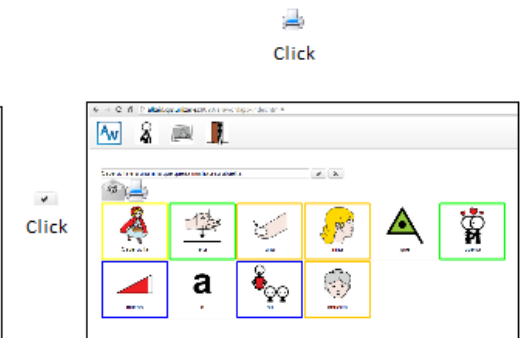
Pantalla de identificación y registro



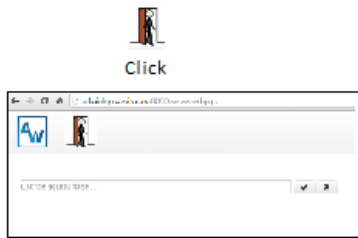
Pantalla inicial de usuario registrado



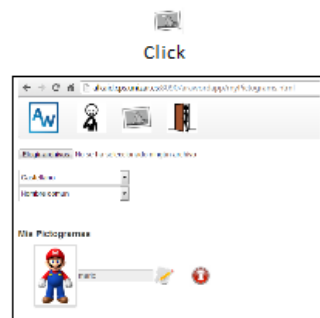
Click en una palabra de un pictograma



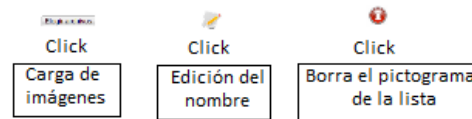
Frase validada con opciones de impresión o enviar por correo electrónico



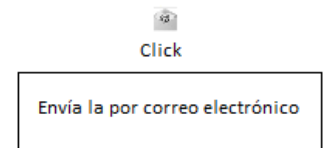
Pantalla inicial con las opciones de escribir texto o de identificarse en la aplicación



Visualización de pictogramas de usuario con opciones de carga, editado y borrado de pictogramas



Cambio de imagen de un pictograma



Click

