

Diseño y desarrollo de una interfaz para la simplificación automática de textos



Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:

Alberto Manuel Cervantes Bañón

Tutor/es:

María Paloma Moreda Pozo



Universitat d'Alacant
Universidad de Alicante

Septiembre 2019

Preámbulo

La actual sociedad de la información obliga a que todos seamos capaces de acceder y comprender la información que Internet nos proporciona. Sin embargo, son muchas las personas que, por muchos motivos, edad, discapacidad, aprendizaje de una lengua etc. no pueden llevar a cabo este acceso de la misma manera que la gran mayoría de las personas. Por ello es necesario disponer de entornos intuitivos y accesibles que permitan superar esta barrera.

Agradecimientos

Quisiera agradecer el apoyo incondicional de todos los que han puesto un poco de ellos en este proyecto, a todas las personas que, con desinterés, me han ayudado y aguantado, sin ellos este proyecto no sería posible.

Por su puesto, agradecer todo el apoyo recibido por parte de mi familia y amigos, esa otra gran familia, gracias por todas las horas de reflexión.

Por último y no por ello menos importante, quiero darle las gracias a Ella, que aun sin “saber lo que hago” ha estado a mi lado durante todo este tiempo, apoyándome y dándome ánimos.

A todos vosotros, muchas gracias.

En el lenguaje científico, la claridad es la única estética permitida.

Gregorio Marañón.

Índice general

1.	Introducción	13
2.	Estado del arte.....	15
2.1.	Simplificar un texto	16
2.1.1.	Lectura fácil en los países nórdicos.....	17
2.1.2.	Marco europeo de personas con discapacidad intelectual	18
2.1.3.	Portal noticias fácil.....	18
2.1.4.	Plena inclusión	19
2.1.5.	Asociación lectura fácil de Barcelona.....	19
2.2.	Simplificación automática de textos.....	20
2.2.1.	Proyecto PSET.....	20
2.2.2.	Proyecto PorSimples.....	21
2.2.3.	Proyecto Simplext.....	21
2.2.4.	Proyecto FIRST	22
2.3.	Conclusiones del Estado del Arte	23
3.	Objetivo	24
3.1.	Objetivos específicos	24
4.	Metodología de desarrollo.....	25
4.1.	Introducción	25
4.2.	Metodologías tradicionales	25
4.2.1.	Microsoft Solution Framework.....	26
4.2.2.	Rational Unified Process (RUP)	27
4.3.	Metodologías ágiles	28
4.3.1.	Extreme Programming (XP).....	29
4.3.2.	Agil Unified Process (AUP)	30
4.3.3.	Scrum	31
4.4.	Conclusión	32
5.	Herramientas de desarrollo.....	33
5.1.	ASP.Net + MVC 5.....	34
5.1.1.	Ventajas	34
5.1.2.	Desventajas.....	34
5.2.	ASP.Net + Vue.js.....	35
5.2.1.	Ventajas	35
5.2.2.	Desventajas.....	35
5.3.	ASP.Net + Razor Pages.....	36
5.3.1.	Ventajas	36
5.3.2.	Desventajas.....	36
5.4.	ASP.Net Core + Blazor.....	37

5.4.1.	Ventajas	37
5.4.2.	Desventajas.....	37
5.5.	Conclusión.....	38
6.	Planteamiento del proyecto	39
6.1.	Casos de uso	40
6.1.1.	Simplificación de textos	40
6.1.2.	Sinónimos	41
6.1.3.	Definiciones.....	42
6.2.	Análisis de requisitos	43
6.3.	Diseño de MockUps	44
7.	Implementación del proyecto.....	45
7.1.	Obtención del corpus de palabras	45
7.2.	Web Scraping.....	47
7.2.1.	El web scraping como sustituto de una API	47
7.3.	Desarrollo de los métodos (Back-End)	48
7.3.1.	Separar frases.....	48
7.3.2.	Limpiar frase.....	48
7.3.3.	Quitar conectores de ejemplificación.....	49
7.3.4.	Obtención de las palabras de la frase.....	49
7.3.5.	Comprobar si la frase está en pasiva.....	50
7.3.6.	Obtener sinónimos	52
7.3.7.	Obtener definiciones	52
7.3.8.	Obtener imágenes palabra.....	52
7.4.	Desarrollo de la interfaz (Front-End)	53
7.4.1.	Pantalla principal	53
7.4.2.	Menú lateral.....	54
7.4.3.	Búsqueda de sinónimos	55
7.4.4.	Búsqueda de definiciones	56
7.4.5.	Simplificar un texto	58
8.	Evaluación y resultados.....	60
8.1.	Quitar texto entre paréntesis y otros símbolos	60
8.2.	Eliminar conectores de ejemplificación	62
8.3.	Cambio de frases voz pasiva a activa.....	63
9.	Conclusiones y trabajo futuro	64
	Bibliografía.....	65

Índice de figuras

Figura 1 Microsoft Solution Framework.....	26
Figura 2 Rational Unified Process (RUP)	27
Figura 3 Extreme Programming (XP)	29
Figura 4 Agile Unified Process (AUP).....	30
Figura 5 Scrum.....	31
Figura 6 Funcionamiento Blazor.....	38
Figura 7 Caso de uso Simplificación de textos	40
Figura 8 Caso de uso de Sinónimos	41
Figura 9 Caso de uso de Definiciones.....	42
Figura 10 MockUp aplicación	44
Figura 11 Interfaz EasyText	53
Figura 12 Menú lateral	54
Figura 13 Hint botón aumentar texto	54
Figura 14 Búsqueda de sinónimos	55
Figura 15 Resultado búsqueda sinónimos	55
Figura 16 Búsqueda de definiciones	56
Figura 17 Resultado de búsqueda de definiciones	56
Figura 18 Definición de los sinónimos.....	57
Figura 19 Sección para simplificar textos.....	58
Figura 20 Sección del texto simplificado.....	58
Figura 21 Resultado de la simplificación	59
Figura 22 Sinónimos/definiciones al pulsar en texto simplificado.....	59

Índice de tablas

Tabla 1 Tabla de ejemplo de etiquetas EAGLE.....	45
Tabla 2 Tabla etiquetas morfológicas adjetivos.....	46
Tabla 3 Ejemplo etiquetas adjetivos	46
Tabla 4 Casos de pruebas para eliminar paréntesis y otros símbolos	60
Tabla 5 Casos de prueba de conectores.....	62
Tabla 6 Caso de pruebas para cambio de voz pasiva a activa.....	63

1. Introducción

El Trastorno del Espectro del Autismo, se define como una disfunción neurológica crónica con fuerte base genética que desde edades tempranas se manifiesta en una serie de síntomas relacionados con la interacción social, la comunicación y la falta de flexibilidad en el razonamiento y comportamientos. Los últimos estudios realizados indican apuntan a una prevalencia de aproximadamente 1 caso de TEA por cada 100 nacimientos (Autism-Europe aisbl 2015).

Las personas con TEA tienen un déficit en la comprensión del lenguaje lo que puede incluir una mala interpretación de los significados no literales o dificultad para comprender instrucciones complejas.

El acceso a la información es un derecho fundamental para todas las personas, por ello, se debe garantizar este derecho para todos los colectivos que bien por tener una discapacidad física como las personas sordas, personas con trastornos del lenguaje como dislexia, personas autistas, personas mayores o simplemente, personas que estén aprendiendo otro idioma, no pueden tener un correcto acceso a la información. Por ello, las Naciones Unidas, en 2006, a través de la *Convención sobre los derechos de las personas*, garantizaron el acceso a la información para dichos colectivos.

En la sociedad actual, existe una tendencia creciente a digitalizar todo tipo de información, desde noticias, documentos oficiales o normativas hasta recetas de cocina, pero, según los estudios realizados, aún estamos lejos de ser una sociedad fuertemente digitalizada, donde la información sea accesible para todos los colectivos.

La información escrita, puede, dependiendo de la forma en la que se presente, excluir a un gran número de personas, las cuales debido a su capacidad lectora les hace tener problemas en la comprensión de ciertos textos. Estas habilidades comprensoras pueden haberse visto afectadas por distintos factores, como, por ejemplo, tener alguna discapacidad cognitiva o haber tenido un acceso a la formación limitado. Por ello, a la hora de presentar la información escrita, debemos tener en cuenta la diversidad de personas que van a tener acceso a ella, para facilitar, en el mayor grado posible, el acceso a la misma.

Nos encontramos, por tanto, ante la problemática de que aun estando en la sociedad en la que se tiene el mayor acceso a la información de la historia, ésta, se presenta de un modo en el que ciertos colectivos de la sociedad se ven incapaces de llegar a comprenderla.

Una primera solución es la simplificación de textos de manera manual y así poder adaptarla, según la dificultad, a los distintos grupos de usuarios a los que va dirigida la información, pero esta tarea, aunque los resultados son satisfactorios, es demasiado lenta y tediosa, por lo que en estos tiempos en los que la información aparece prácticamente a cada segundo en la red esta solución no es viable.

La simplificación automática de textos dentro del procesamiento del lenguaje natural es una disciplina relativamente nueva, el objetivo, es realizar mediante procesos automáticos la transformación de un texto en otro equivalente que sea más fácil de entender para los usuarios de dichos textos.

2. Estado del arte

Se calcula que un 30% de la población tiene dificultades lectoras, las cuales pueden deberse a distintos factores, pero indistintamente de la causa de las dificultades hacen que se necesite una versión simplificada de los textos para que estas personas, puedan tener un acceso libre a la información.

A continuación, se va a realizar una revisión de las propuestas de simplificación manual desarrolladas hasta hoy y de las aproximaciones de simplificación automática que se han implementado en esta área.

2.1. Simplificar un texto

La necesidad de simplificar un texto surge debido a los problemas que aparecen al acceder a la información que nos llega, principalmente a la escrita. Nos encontramos con muchos problemas a la hora de la lectura y comprensión del texto, además, el uso de palabras poco frecuentes, de abreviaturas o de palabras en otras lenguas, no facilitan la comprensión.

La complejidad de un texto se puede medir a partir de tres factores: la complejidad de los segmentos (naturaleza y longitud), la complejidad léxica (frecuencia y densidad léxica) y la complejidad estructural de las oraciones factores (Actas del Sexto Congreso de Hispanistas de Asia, 2008).

Controlando estos factores de complejidad es posible diseñar estrategias de simplificación de textos para garantizar una comprensión correcta, con el estudio de estos factores, podemos saber que la frecuencia de uso de las palabras condiciona la comprensión de un texto, es decir, cuantas más palabras poco utilizadas en el lenguaje, más difícil será la comprensión del texto que las contenga. Por otra parte, los textos que contienen más de una oración y con segmentos de texto más largos, presentan una mayor complejidad para el lector, por lo que hará que la comprensión del texto por parte de este sea más complicada.

Desde el punto de vista computacional, existen distintas métricas para medir o cuantificar lo sencillo o complejo que es la lectura de un texto, estas métricas, basadas en ecuaciones matemáticas miden, desde la cantidad de pronombres que nos encontramos en el texto hasta el número de sílabas por palabras. Estas métricas se encuentran disponibles tanto para el inglés¹ como para el castellano (Rodríguez Diéguez, y otros, 1993).

Existen ciertas iniciativas que proponen pautas para ayudar a reescribir un texto de manera más comprensible. Algunas de ellas, por ejemplo, son las *Directrices Europeas para generar Información de Fácil Lectura* (Freyhoff, y otros, 1998), *Plain language*² o las últimas guías para el *Contenido Accesible en la web (WCAG 2.0)*³.

Al ritmo que cambian los contenidos digitales en nuestra sociedad, es muy complicado mantenerse al día con todos los medios de noticias o redes sociales que a diario inundan nuestra vida. Debido a esto, se han creado contenidos simplificados manualmente a partir de

¹ <https://w-shadow.com/blog/2009/04/28/calculating-readability-metrics-in-php/>

² <https://www.plainlanguage.gov/>

³ <https://www.w3.org/TR/WCAG/>

material ya existente, como es el caso de *Simple English Wikipedia*⁴ donde se han simplificado más de 100.000 artículos sacados de la *Wikipedia*, o el portal de *Noticias Fáciles* en español de *Discapnet*⁵, donde de manera manual, se adaptan diariamente noticias de prensa.

Siguiendo la misma línea, existe un portal, promovido por la Unión Europea en la que se incluyen más de 60 organizaciones de personas con discapacidad intelectual de toda Europa en la que todas estas personas, pueden acceder a la información de una manera simple y fácil de comprender, se trata del proyecto *Inclusion Europe*.

2.1.1. Lectura fácil en los países nórdicos

A finales de la década de los sesenta (1968) surgió una iniciativa en Suecia para la adaptación de textos a lectura fácil, cuyo resultado, el cual pertenece al Ministerio de Cultura es el actual centro de lectura fácil *Myndigheten För Tillgängliga Medier*⁶, después de varias décadas, en el año 1984, se lanzó el primer periódico en lectura fácil, titulado *8Sidor*⁷ pero empezó a publicarse de forma permanente en 1987.

En Noruega, se lanzó la una iniciativa llamada Lector busca libro (*Leser Soker bok*) que se fundó gracias a la colaboración de más de 20 organizaciones, entre las que se incluye una editorial y varias organizaciones de personas con discapacidad. Desde su creación en 2003 ha editado más de 50 títulos.

En Finlandia existen dos centros, ambos vinculados a organizaciones de personas con discapacidad intelectual. Ambos centros publican seminarios, folletos de información y libros, todos ellos en lectura fácil.

⁴ https://simple.wikipedia.org/wiki/Main_Page

⁵ <http://www.noticiasfacil.es/ES/Paginas/index.aspx>

⁶ <https://www.mtm.se>

⁷ <https://8sidor.se/>

2.1.2. Marco europeo de personas con discapacidad intelectual

*Inclusion Europe*⁸ se trata de una organización que se creó en 1988 con sede en Bruselas cuyo objetivo es luchar por la igualdad de derechos y plena inclusión de personas con discapacidad intelectual en todos los aspectos de sus vidas, por ello se trata del punto de encuentro de las asociaciones de personas con discapacidad intelectual en la Unión Europea.

Elaboró la guía *El camino más fácil: Directrices europeas para generar información de fácil lectura destinada a personas con discapacidad intelectual*, en la que se proponen las pautas para desarrollar un proyecto de redacción en lectura fácil.

2.1.3. Portal noticias fácil

El portal de noticias web *Noticias fácil*⁹ se creó desde la fundación ONCE¹⁰ y se dirige a personas que ya sea por una discapacidad intelectual o cognitiva tienen problemas a la hora de comprender y asimilar textos. Publica noticias, libros y documentos en lectura fácil, lo que facilita que la información sea accesible a todas las personas.

Otra de sus pretensiones es la de ayudar en la mejora y de las habilidades de comprensión y de comunicación de las personas con discapacidad intelectual, así como la de ser un punto de encuentro a través de encuestas y blogs.

⁸ <https://www.inclusion-europe.eu/>

⁹ <http://www.noticiasfacil.es/ES/Paginas/index.aspx>

¹⁰ <https://www.fundaciononce.es/>

2.1.4. Plena inclusión

*Plena inclusión*¹¹ se trata de una organización que representa en España, a las personas con discapacidad intelectual o del desarrollo. Lleva más de 50 años contribuyendo en la plena inclusión de estas personas y de sus familias en la sociedad.

Está formada por 19 federaciones autonómicas, incluyendo Ceuta y Melilla y casi 900 asociaciones de toda España, defendiendo los derechos de estas personas para que puedan participar en la sociedad al igual que el resto de los ciudadanos.

2.1.5. Asociación lectura fácil de Barcelona

*La Asociación Lectura Fácil de Barcelona*¹² fue una de las primeras asociaciones en España de estas características. Desde que fue fundada en 2002 trabaja para acercar la lectura a personas con dificultades lectoras y para ello cuenta con más de 120 libros de lectura fácil y casi un centenar de clubes de lectura fácil asociados que promueven la lectura fácil.

Ha asesorado a entidades públicas para adaptar folletos y textos informativos a lectura fácil, partiendo del principio de democracia lectora y es que todo el mundo debe tener acceso a la información, literatura y cultura, para poder participar de forma responsable y activa en la sociedad.

¹¹ <http://www.plenainclusion.org>

¹² <http://www.lecturafacil.net/>

2.2. Simplificación automática de textos

En los proyectos e iniciativas vistas anteriormente la adaptación de textos a lectura fácil o textos de mayor comprensión, se realizan de manera manual. Pero esta tarea de simplificación es una tarea compleja y conlleva un trabajo que tanto en tiempo como en recursos es muy costoso. Actualmente, la información se genera de una manera tan rápida que es imposible realizar una adaptación de todo este contenido de manera manual y que sea accesible en tiempo real. Es la simplificación automática de textos la solución a esta problemática.

Los primeros estudios de la simplificación automática de textos cubrían mucho terreno, los sistemas que aprenden reglas de simplificación, el análisis de las cuestiones de simplificación sintáctica y léxica, coherencia del texto, etc. Con el paso del tiempo, muchas de estas ideas han sido olvidadas, sin embargo, otras han sido redescubiertas, reforzando estos estudios.

2.2.1. Proyecto PSET

El proyecto *Practical Simplification of English Text* (Carroll, y otros, 1998) fue uno de los primeros en aplicar tecnologías de lenguaje natural para personas con dificultades lectoras, tenía como objetivo simplificar las noticias de prensa en inglés para personas con afasia. Para ello se utilizaba la simplificación sintáctica, resolución de anáforas y simplificación léxica

Seleccionaba cada palabra y con ella, creaba un archivo de texto con todos los sinónimos de las palabras, una vez creado este archivo, seleccionaba la palabra adecuada según el criterio de mayor frecuencia, para llevar a cabo esta tarea, el proyecto se nutría de la base de datos psicolingüística desarrollada en Oxford, *The Oxford Psycholinguistic Database* (Quinlan, 1992).

2.2.2. Proyecto PorSimples

Se trata de un sistema, desarrollado para ayudar a lectores con baja alfabetización a procesar de manera correcta documentos de la web, para el idioma portugués. Este sistema fue presentado gracias a los trabajos de Aluísio (Aluísio, y otros) y Candido (Candido, y otros).

A la hora de realizar el proyecto utilizaron diferentes técnicas de adaptación de textos, como resumen automático para hacer que los textos más cortos, simplificación léxica para realizar la sustitución de palabras complejas o de menor uso por otras más simples o de mayor uso, simplificación sintáctica para separar oraciones complejas.

2.2.3. Proyecto Simplext

El proyecto Simplext¹³ (Sistema automático de transformación de contenidos en textos de fácil lectura) tiene como objetivo principal el desarrollo un producto para la simplificación automática de textos para el español dirigido a personas con discapacidad intelectual, financiado por el Ministerio de Industria, Energía y Turismo, mediante el Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica (I+D+I), Plan Avanza 2.

Este proyecto permite facilitar la lectura de textos en base a la reducción de la complejidad tanto léxica como sintáctica de los textos, detectando las palabras y oraciones más complejas y generando vocabulario más simple y oraciones más cortas lo que facilita la comprensión del texto.

¹³ <http://simplext.taln.upf.edu/>

2.2.4. Proyecto FIRST

El proyecto europeo *FIRST*¹⁴ (Flexible Interactive Reading Support Tool) está orientado a desarrollar una herramienta llamada *Open Book*¹⁵ que utiliza tecnologías del lenguaje humano para identificar obstáculos que dificultan la comprensión lectura de los documentos. La herramienta pretende ayudar tanto a los cuidadores como a las personas con autismo, transformando textos a otros más sencillos eliminando los obstáculos identificados en el texto, como conceptos difíciles, metáforas, etc...

Con este proyecto se espera que mejore la calidad de vida de las personas con TEA (Trastorno del Espectro del Autismo), así como la mejora en su acceso a la educación y acceso con mayores oportunidades al ámbito profesional.

¹⁴ <http://www.first-asd.eu/?q=about>

¹⁵ <http://www.openbooktool.net/>

2.3. Conclusiones del Estado del Arte

Se ha mostrado la tarea de simplificar un texto, ya sea de manera manual o de manera automática, identificando las principales operaciones en el proceso de simplificación, mostrando la necesidad de realizar esta tarea.

También se han revisados distintas propuestas de simplificación manual desarrolladas tanto por iniciativas nacionales y europeas. Además, se han revisado distintas propuestas y proyectos de simplificación automática de textos.

Los proyectos existentes actualmente, realizan una simplificación basada en separar las frases en frases más cortas, en sustituir, directamente palabras complejas por su sinónimo y en la eliminación de los paréntesis del texto. Bajo mi punto de vista, esta característica, aunque excelente, se puede mejorar, ampliando los recursos que la aplicación puede ofrecer.

Creo que se debería de añadir secciones para buscar sinónimos sin necesidad de introducir todo un texto o buscar definiciones para palabras que puedan ser complejas para el usuario.

En cuanto al resumen, lo haría más rico, resaltando las palabras complejas y permitiendo ver de un vistazo sus sinónimos más frecuentes o su significado, para poder aplicarlo al contexto del texto simplificado.

Otra característica que se echa en falta es poder cambiar los tamaños de los textos, para personas que tienen dificultades lectoras o cambiar los estilos de la página/textos para adaptarlos a personas con daltonismo.

3. Objetivo

El acceso a la información accesible para todos es el interés general de este trabajo y para ello propondremos un modelo genérico para realizar el proceso automático de simplificación de textos. El objetivo es hacer la información más accesible, reescribiendo los textos complejos, para que estos sean más comprensibles. Por lo tanto, propondremos ciertas etapas específicas dentro del modelo genérico, para poder llevar a cabo esta tarea. Esto requiere cierto conjunto de estrategias de reescritura que produzcan expresiones más simples, más fáciles de entender que las frases originales, pero siempre, cercanas al sentido original.

3.1. Objetivos específicos

Existen cuatro grandes tareas que, durante los años de investigación sobre la simplificación de textos, todos los trabajos desarrollados han tratado de una manera u otra, estas son:

1. Simplificación sintáctica: Transformar oraciones largas y complejas en otras oraciones más simples e independientes, cambiando oraciones de pasiva a activa, segmentando oraciones subordinadas y coordinadas.
2. Enriquecimiento léxico: Ayudar a la comprensión del vocabulario complejo, pudiendo ver los sinónimos de las palabras complejas.
3. Eliminar información: Debemos prescindir de la información redundante para ayudar a la comprensión del texto, eliminando por ejemplo texto entre paréntesis.
4. Clasificación de información: A los conceptos que se consideren complejos, se le añadirán explicaciones, por lo que hay que definir que conceptos son difíciles pero importantes, por lo que no se deben eliminar, sino añadir una definición.

4. Metodología de desarrollo

4.1. Introducción

Una metodología de desarrollo de software es un marco de trabajo, el cual es usado para planear, controlar y estructurar el proceso de desarrollo en sistemas de información, es decir, se trata de una metodología con un enfoque estructurado y estratégico que permite el desarrollo de software en base a un modelo de sistema, reglas y guías.

4.2. Metodologías tradicionales

Al inicio el desarrollo de software era artesanal en su totalidad, la fuerte necesidad de mejorar el proceso y llevar los proyectos a la meta deseada, tuvieron que importarse la concepción y fundamentos de metodologías existentes en otras áreas y adaptarlas al desarrollo de software.

Esta nueva etapa de adaptación contenía el desarrollo dividido en etapas de manera secuencial que de algo mejoraba la necesidad latente en el campo del software.

Entre las principales metodologías tradicionales tenemos los ya tan conocidos RUP y MSF entre otros, que centran su atención en llevar una documentación exhaustiva de todo el proyecto y centran su atención en cumplir con un plan de proyecto.

Otra de las características importantes dentro de este enfoque tenemos los altos costos al implementar un cambio y al no ofrecer una buena solución para proyectos donde el entorno es volátil. Las metodologías tradicionales (formales) se focalizan en documentación, planificación y procesos.

Estas metodologías se centran en llevar una documentación exhaustiva de todo el proceso de desarrollo software y en desarrollar un gran plan de proyecto que está definido desde la fase inicial del mismo.

4.2.1. Microsoft Solution Framework

Metodología creada por Microsoft Consulting Services que trata de definir un marco de trabajo para construir e implantar sistemas distribuidos basados en herramientas y tecnologías de Microsoft en cualquier plataforma.

Consta de cinco fases diferenciadas:

1. Visión
2. Planeación
3. Desarrollo
4. Implantación
5. Soporte

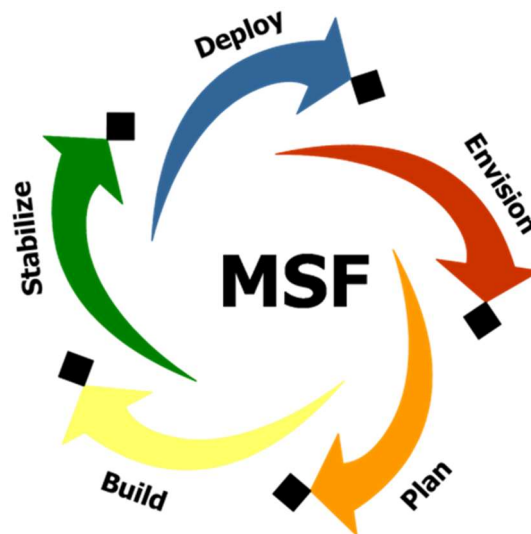


Figura 1 Microsoft Solution Framework

4.2.2. Rational Unified Process (RUP)

Se trata de un modelo de desarrollo de proyectos de software el cual define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto. Tiene como objetivo asegurar que la producción de software sea de alta calidad y que satisfaga los requerimientos de los usuarios finales. Utiliza UML¹⁶ como lenguaje de notación.

Podemos observar que RUP, consta de cuatro fases en su ciclo de vida:

1. Concepción
2. Elaboración
3. Construcción
4. Transición

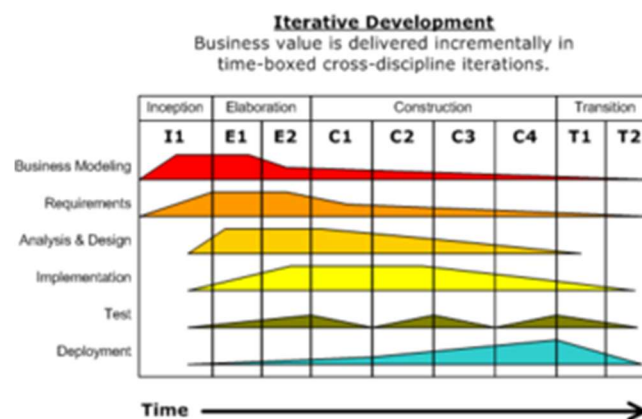


Figura 2 Rational Unified Process (RUP)

¹⁶ https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado

4.3. Metodologías ágiles

Luego de varias opiniones tanto a favor como en contra de las metodologías tradicionales se genera un nuevo enfoque denominado, métodos ágiles, que nace como respuesta a los problemas detallados anteriormente y se basa en dos aspectos puntuales, el retrasar las decisiones y la planificación adaptativa, permitiendo potencia aún más el desarrollo de software a gran escala.

Como resultado de esta nueva teoría se crea un Manifiesto Ágil¹⁷ cuyas principales ideas son:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados.
- Es más importante crear un producto software que funcione que escribir documentación exhaustiva.
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Entre los principales métodos ágiles tenemos el XP (eXtreme Programming), Scrum, AUP entre otras.

Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. Nos lo proponen porque para muchos clientes esta flexibilidad será una ventaja competitiva y porque estar preparados para el cambio significa reducir su coste.

¹⁷ <https://agilemanifesto.org/iso/es/principles.html>

4.3.1. Extreme Programming (XP)

Metodología ágil basada en la realimentación continua entre el cliente y el equipo de desarrollo, gracias a una comunicación fluida entre todos los participantes. Es especialmente adecuada para los proyectos con requisitos no muy bien definidos o cambiantes ya que se centra más en la adaptabilidad que en la previsibilidad.

Los defensores de XP consideran que los cambios en los requisitos son un aspecto natural e inevitable en cualquier desarrollo, por lo que ser capaz de adaptarse a ellos es una mejor aproximación que intentar definir, desde el principio del proyecto, todos los requisitos.

- Desarrollo iterativo e incremental
- Pruebas unitarias continuas
- Programación por parejas
- Frecuente iteración
- Corrección de errores
- Propiedad del código compartida
- Simplicidad en el código

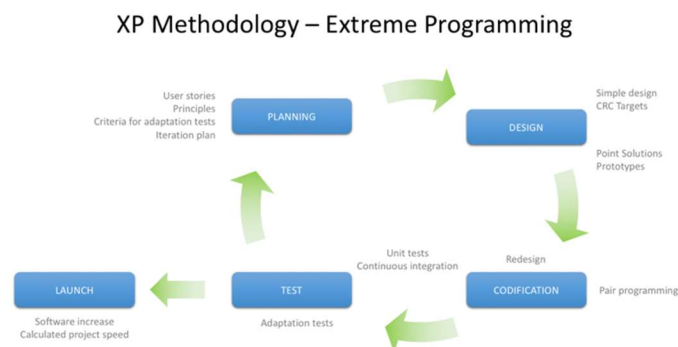


Figura 3 Extreme Programming (XP)

4.3.2. Agil Unified Process (AUP)

Se trata de una metodología basada en la disciplina y las entregas incrementales en el tiempo. Dependiendo del tipo de proyecto, el ciclo de vida de esta metodología puede variar, siendo para proyectos grandes serial, pero para proyectos de menor tamaño, iterativo.

Las características fundamentales de AUP son:

- Modelado
- Implementación
- Prueba
- Despliegue
- Administración de la configuración
- Administración o gerencia del proyecto
- Entorno

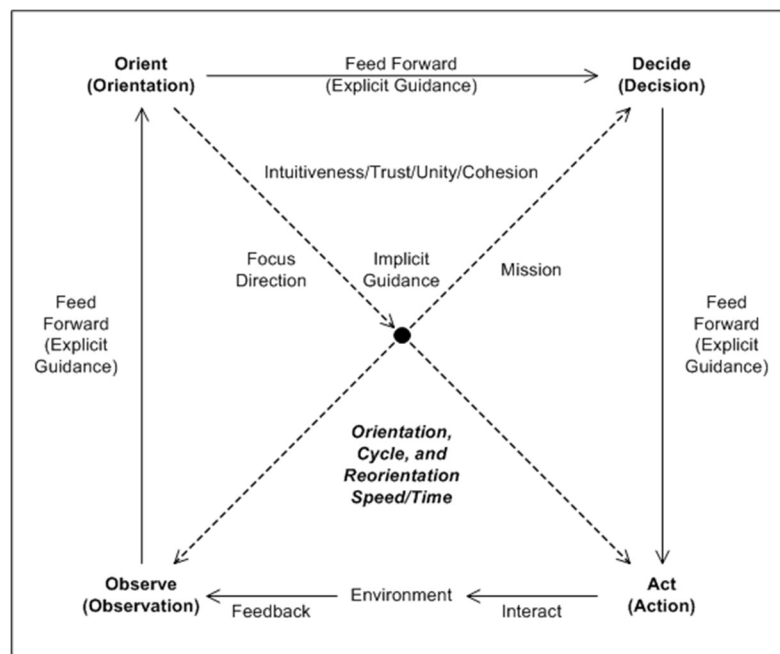


Figura 4 Agile Unified Process (AUP)

4.3.3. Scrum

Se trata de una metodología ágil, la cual sirve para administrar y controlar el desarrollo de software. Este desarrollo comienza de una manera iterativa e incremental, cada ciclo de la iteración termina con una parte del software ejecutable, que añade una nueva funcionalidad al conjunto del sistema software.

Está diseñado para ser adaptativo, por lo que los cambios en los requerimientos software no le afectan ya que estos requerimientos se revisan y ajustan en intervalos muy cortos y regulares durante la construcción del proyecto. Lo que se busca con la metodología Scrum es que los equipos de desarrollo sean lo más efectivos y productivos posible.

Con esta metodología, el cliente se compromete con el proyecto ya que va viendo como crece el producto en cada iteración, por su parte, el equipo de desarrollo incrementa su motivación ya que gracias a esta metodología encuentran un ámbito adecuado para desarrollar sus capacidades.

Scrum cuenta con un sencillo conjunto de reglas:

- Inspección continua
- Adaptación
- Autogestión
- Innovación

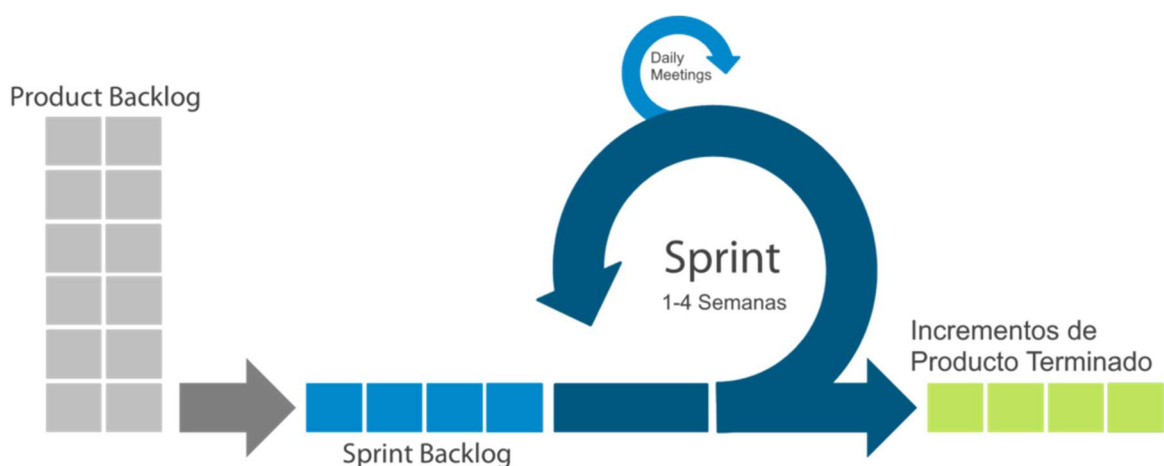


Figura 5 Scrum

4.4. Conclusión

El retrasar las decisiones en un proyecto de software permite potenciar el valor del producto tanto para el cliente como al equipo o empresa que desarrolla.

Para que un grupo de desarrollo adopte una metodología ágil debe poseer experiencia trabajando con metodologías tradicionales, ya que la experiencia es la que predomina en los momentos cruciales del proyecto, además debe tener la capacidad de ser equipos auto-gestionados, altamente motivados y con gran innovación.

Las metodologías ágiles permiten disminuir costos y brindar flexibilidad a los proyectos de software donde la incertidumbre está presente.

Las metodologías ágiles se deberían aplicar en proyectos donde exista mucha incertidumbre donde el entorno es volátil, donde los requisitos no se conocen con exactitud, mientras que las metodologías tradicionales obligan al cliente a tomar las decisiones al inicio del proyecto.

Debido a que las modificaciones en los requisitos son algo característico de los proyectos actuales y dado que, en el tiempo de la inmediatez en el que vivimos, que el cliente tenga una actualización del software cada poco tiempo para que valide si se está trabajando en el camino correcto es vital, por lo que se ha optado por el uso de una metodología ágil y dentro de este tipo de metodologías, se ha optado por usar la metodología Scrum.

5. Herramientas de desarrollo

Para el desarrollo de la aplicación para la simplificación de textos automáticos podemos utilizar diversas herramientas de desarrollo, por ello, expondremos las más significativas según nuestro criterio y evaluaremos sus ventajas y desventajas, para poder seleccionar la herramienta que mejor se adapte tanto a las necesidades del proyecto como al equipo de desarrollo.

Todas las herramientas de desarrollo que veremos a continuación son herramientas que están enmarcadas en el ecosistema de Microsoft. Este ecosistema tiene una de las comunidades más grandes y activas, por lo que podremos valernos de ella para solventar los problemas que vayan surgiendo durante el desarrollo de la aplicación.

Por otro lado, se han seleccionado las herramientas del ecosistema de Microsoft debido al giro que esta empresa ha realizado en cuanto al desarrollo de aplicaciones móviles, dejando atrás el desarrollo para Windows Phone y se ha centrado en el desarrollo de aplicaciones para Android.

5.1. ASP.Net + MVC 5

MVC (Modelo-Vista-Controlador) es un patrón de arquitectura que permite a los desarrolladores de software construir una aplicación web como la conjunción de tres funciones: modelo, vista y controlador.

- Modelo: Gestiona los datos de la aplicación.
- Vista: Contiene la representación visual de la aplicación.
- Controlador: Es el intermediario entre el modelo y la vista.

5.1.1. Ventajas

1. Debido a la separación entre capas, el desarrollo en paralelo es más sencillo, ya que un programador puede estar trabajando en las vistas y otro en la gestión de los datos.
2. Se pueden crear distintas vistas para un mismo modelo, limitando de esta manera la duplicidad del código.
3. La arquitectura MVC puede integrarse fácilmente con el Framework de JavaScript.
4. Debido a que el patrón MVC devuelve los datos sin aplicar ningún formato, los mismos componentes pueden utilizarse con cualquier interfaz.

5.1.2. Desventajas

1. La complejidad de desarrollo es alta al utilizar el patrón MVC.
2. No es adecuado para aplicaciones pequeñas ya que tiene un efecto adverso en el rendimiento.
3. Si diferentes equipos están trabajando sobre las clases de modelo, estos deben ir al mismo ritmo de desarrollo.
4. Alta dependencia entre la vista y el modelo con el controlador.

5.2. ASP.Net + Vue.js

Se trata de un marco de código abierto de JavaScript para la creación de interfaces de usuario y aplicaciones del tipo SPA (Simple Page Application). Se centra en la representación declarativa y la creación de componentes.

5.2.1. Ventajas

1. Tamaño: La compilación de Vue.js apenas llega a pesar 18kb.
2. Facilidad de comprensión y desarrollo ya que Vue.js permite el desarrollo de plantillas a gran escala.
3. Fácil integración ya que es útil tanto para crear aplicaciones completas como para crear componentes para aplicaciones existentes.
4. La documentación de Vue.js es muy completa.
5. Flexibilidad.

5.2.2. Desventajas

1. Falta de algunos componentes sólidos comunes que facilitan el trabajo.
2. Debido a la gran evolución que está teniendo Vue.js los ejemplos que podemos llegar a encontrar quedan obsoletos rápidamente, lo que puede causar cierta frustración en el desarrollador.
3. Se han encontrado problemas menores con iOS y Safari, sobre todo si estos son de versiones algo anticuadas.

5.3. ASP.Net + Razor Pages

Las páginas de Razor son una nueva característica de ASP.NET Core MVC, el cual facilita la codificación de escenarios centrados en páginas, lo que hace que sea más productiva.

5.3.1. Ventajas

1. Provee una sintaxis optimizada que permite una transición mínima entre el HTML y el código.
2. Permite un flujo de trabajo más fluido de codificación.
3. Minimiza el número de caracteres y las pulsaciones, haciendo que la sintaxis sea más compacta y fácil escribir el código.
4. Facilidad de aprendizaje.

5.3.2. Desventajas

1. La sintaxis está muy orientada a HTML, el modelo de datos de Razor es esencialmente una concatenación de cadenas, por lo que los errores de sistaxis y anidación no se detectan ni de forma estática ni dinámica.
2. Razor confunde el código del servidor y el HTML/JavaScript, lo que hace que el desarrollo tanto en HTML como en JS sea en ocasiones un desafío.
3. Mala encapsulación y capacidad de reutilización.

5.4. ASP.Net Core + Blazor

Se trata de una interfaz de usuario, diseñado para simplificar la tarea de crear aplicaciones de tipo SPA (Simple Page Application) de manera rápida, pudiéndose ejecutar en cualquier navegador.

5.4.1. Ventajas

1. Dado que la actualización de la interfaz de usuario se maneja a través de una conexión SignalR, podemos evitar actualizaciones innecesarias de la página.
2. El tamaño de descarga de la aplicación es muy pequeño, por lo que la carga inicial es más rápida.
3. El componente de Blazor, puede aprovechar al máximo las capacidades del servidor, así como el uso de API's.
4. Dado que Blazor en el lado del servidor se ejecuta bajo un proceso nativo de .NET Core, es compatible con los navegadores que no tienen soporte para WebAssembly.

5.4.2. Desventajas

1. La interacción con el interfaz de usuario implica la comunicación de SignalR, agrega un paso adicional en las llamadas de red, lo que provoca cierta latencia.
2. La escalabilidad de las aplicaciones, en ocasiones puede suponer un desafío.

5.5. Conclusión

Teniendo en cuenta que al trabajar con Blazor dejamos de lado la utilización de JavaScript, aunque se mantiene la interoperabilidad entre ambos, podemos desarrollar nuestro proyecto bajo un lenguaje tan potente como es C#, lo que facilita el desarrollo de este ya que se trata de un lenguaje muy familiar para nuestro desarrollo.

Por otro lado, ASP.Net Core es un potente marco de programación para el desarrollo web del lado del servidor, por lo que al habilitar C# en el lado del cliente permitirá el uso de una tecnología común tanto en el servidor como en el cliente.

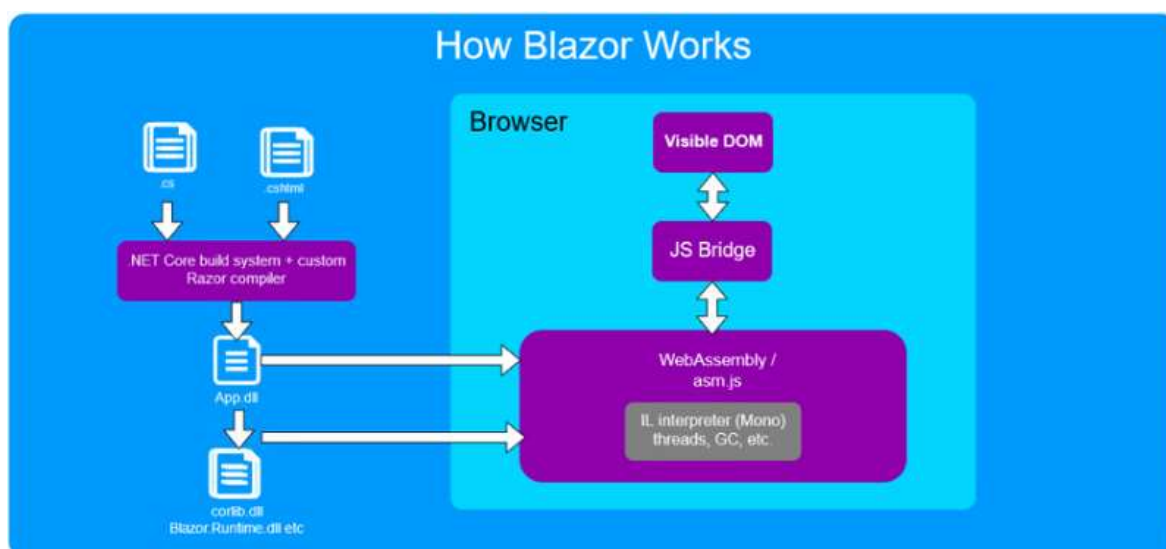


Figura 6 Funcionamiento Blazor

6. Planteamiento del proyecto

Como se ha comentado con anterioridad, se va a desarrollar una aplicación web, que se centrará en la simplificación automática de textos, aunque tendrá otros servicios como, por ejemplo, la obtención de sinónimos/definiciones de palabras o cambio del tamaño de los textos para adaptarlo a personas con visión reducida.

- **Sinónimos:** Devuelve una lista de todos los sinónimos de la palabra seleccionada/introducida
- **Definiciones:** Devuelve la definición de la palabra que se desea buscar, mostrando su definición, mostrando, además una imagen para ayudar a su comprensión.
- **Simplificación de Texto:** Devuelve el texto simplificado y enriquecido.

Ayudándonos de estos servicios, realizaremos la simplificación del texto, pudiendo ver los sinónimos y las definiciones de todas las palabras del texto simplificado.

6.1. Casos de uso

Como ya se ha comentado, existen tres grandes casos de uso en la aplicación, los cuales pasaremos a ver en detalle.

6.1.1. Simplificación de textos

El principal caso de uso de la aplicación es la simplificación de textos, en esta sección, el usuario podrá introducir un texto y cuando use la opción de “simplificar” se le devolverá el texto simplificado y enriquecido, ya que podrá obtener la definición y los sinónimos de las palabras que no comprenda.

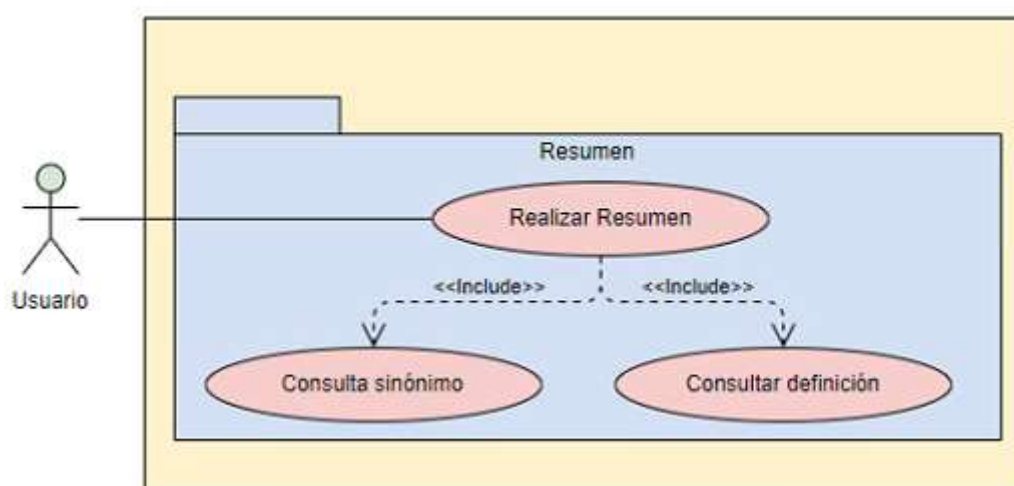


Figura 7 Caso de uso Simplificación de textos

6.1.2. Sinónimos

La búsqueda de sinónimos se podrá realizar independientemente de si se ha querido realizar un resumen de un texto o no, en este caso de uso, el usuario introducirá una palabra de la cual quiere saber sus sinónimos y la aplicación le devolverá un listado de los sinónimos más frecuentes, pudiendo, marcando en uno de ellos ver su definición.

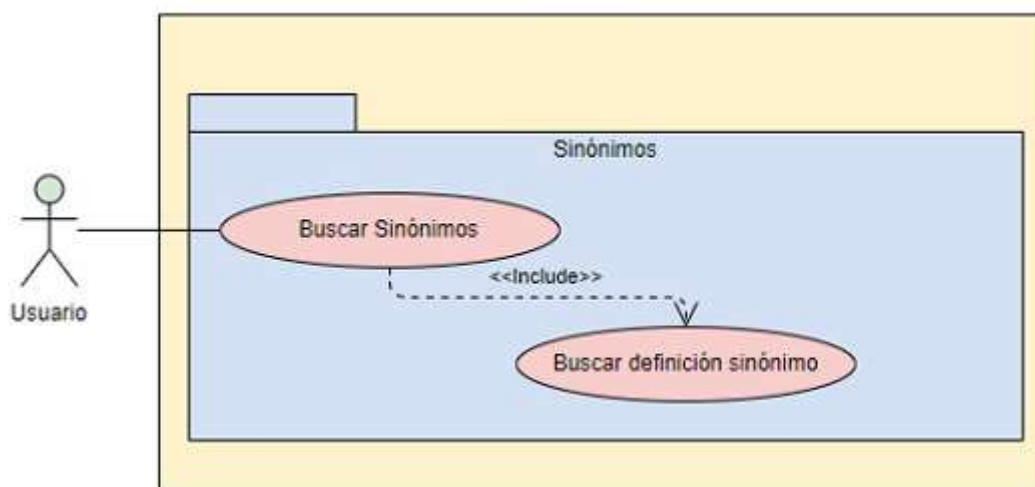


Figura 8 Caso de uso de Sinónimos

6.1.3. Definiciones

La obtención de definiciones de palabras será el tercer gran caso de uso de la aplicación, el usuario introducirá una palabra en el apartado de definiciones y obtendrá la definición de esta, además se mostrará una imagen que puede ayudar a la comprensión de esta.

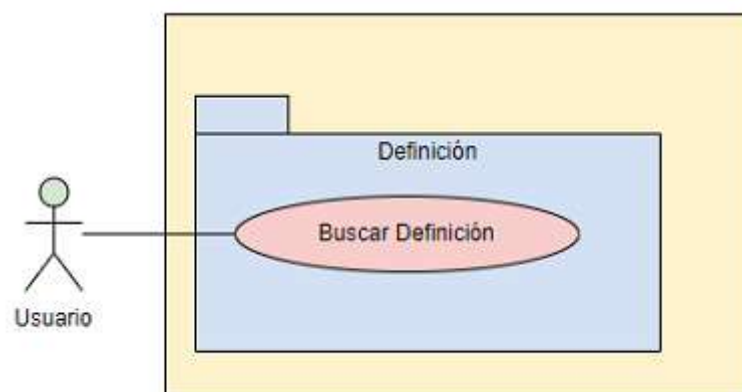


Figura 9 Caso de uso de Definiciones

6.2. Análisis de requisitos

Basándonos en los requisitos de Lectura fácil¹⁸ marcados por la Unión Europea, realizaremos los máximos cambios posibles en el texto original para que la simplificación que realicemos del mismo se adapte en la mayor medida posible a sus requisitos.

Tanto para la simplificación de textos, como para la búsqueda de sinónimos o de las definiciones nos basaremos en las reglas indicadas por la Unión Europea:

- Evitaremos el uso de palabras difíciles. Si es necesario su uso, se podrá ver una explicación de esta para facilitar su comprensión.
- Evitar el uso de iniciales, por ejemplo, sustituir “UE” por “Unión Europea”.
- Hacer siempre frases cortas.
- Utilizar frases en activa en lugar de frases en pasiva.
- No usar fondos debajo de los textos para facilitar su lectura.
- Utilizar una tipología de letra clara y fácil de leer, como, por ejemplo, Aria o Tahoma.
- Evitar el uso de cursivas o subrayado de palabras.
- Evitar caracteres especiales, como “\, &, <, >, #”
- Evitar las abreviaturas como “p.e.” o “etc.”
- Empezar una frase nueva en una línea nueva.

¹⁸ <https://easy-to-read.eu/es/european-standards/>

6.3. Diseño de MockUps

Una vez realizado los casos de uso de la aplicación y haber obtenido los requisitos de esta, pasaremos al diseño en mockups de esta, para hacernos una idea de cómo queremos mostrar estos casos de uso en la aplicación.

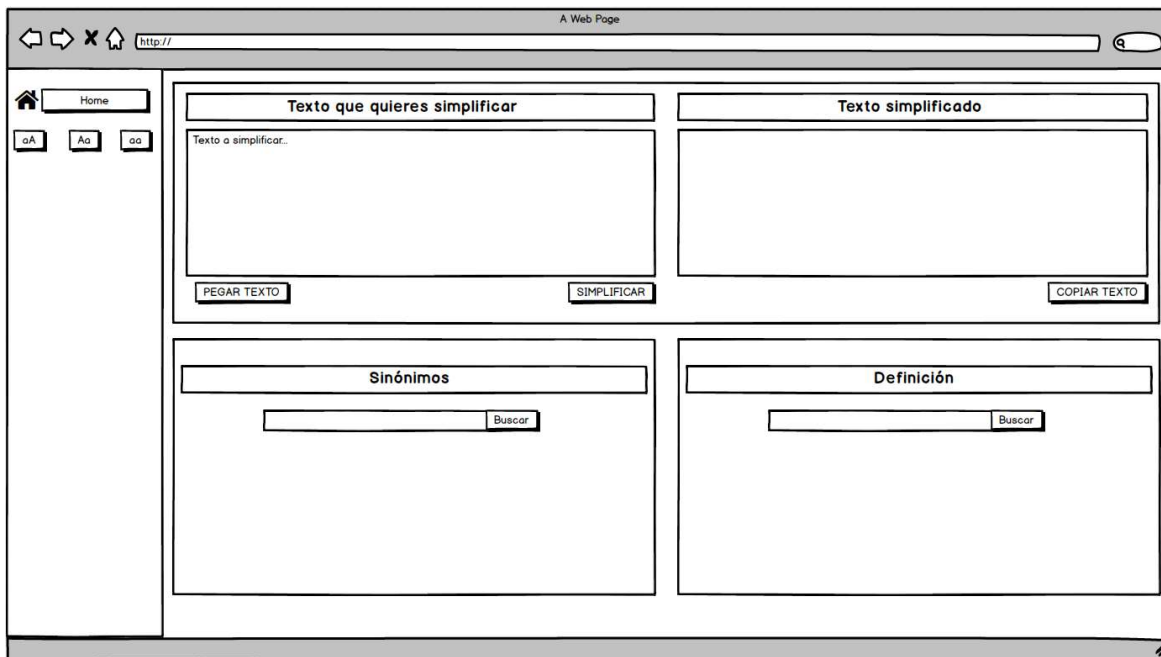


Figura 10 MockUp aplicación

7. Implementación del proyecto

7.1. Obtención del corpus de palabras

Para poder realizar ciertas tareas en la simplificación de los textos, como, por ejemplo, el paso de las frases que estén en pasiva a activa, es necesario saber el tipo de palabras que componen la frase, es decir, saber que palabra es un verbo o un sustantivo, pasando por adjetivos, adverbios, etc.

Estos listados de palabras están acompañados de un conjunto de etiquetas que representan la información morfológica de las palabras. Este conjunto de etiquetas se basa en las etiquetas propuestas por el grupo EAGLE¹⁹ para la anotación morfosintáctica de lexicones y corpus para todas las lenguas europeas. Por ello, dependiendo de la lengua, hay atributos que pueden no especificarse, esta infraespecificación de un atributo, se marca con el valor 0.

A continuación, se muestra un ejemplo de las etiquetas que el analizador morfológico utiliza para el castellano en formato tabla.

ETIQUETAS			
Posición	Atributo	Valor	Código
<i>Columna 1</i>	<i>Columna 2</i>	<i>Columna 3</i>	<i>Columna 4</i>

Tabla 1 Tabla de ejemplo de etiquetas EAGLE

En la *columna 1* encontramos el número que hace referencia al orden y posición en que aparecen los atributos. La *columna 2* hace referencia a los atributos, que, dependiendo de la categoría, pueden variar. En la *columna 3* nos encontramos los valores que puede tomar cada atributo de la *columna 2* y, por último, en la *columna 4* se representan los códigos que se han establecido para su representación.

¹⁹ <https://www.cs.upc.edu/~nlp/tools/parole-sp.html>

Paso a mostrar un pequeño ejemplo utilizaré los adjetivos, para tener una mejor visión de cómo se representan morfológicamente las palabras.

ADJETIVOS			
Posición	Atributo	Valor	Código
1	Categoría	Adjetivo	A
2	Tipo	Calificativo	Q
3	Grado	Apreciativo	A
4	Género	Masculino	M
		Femenino	F
		Común	C
5	Número	Singular	S
		Plural	P
		Invariable	N
6	Caso	-	0
7	Función	Participio	P

Tabla 2 Tabla etiquetas morfológicas adjetivos

Una vez tenemos las posiciones y lo que representa cada una de ellas, pasemos a un breve ejemplo con unos adjetivos.

Forma	Etiqueta
Alegres	AQ0CP00
Alegre	AQ0CS00
Bonitas	AQ0FP00
Bonita	AQ0FS00
Bonitos	AQ0MP00
Bonito	AQ0MS00
Quemada	AQ0FS0P

Tabla 3 Ejemplo etiquetas adjetivos

Para obtener el listado del resto de tipos de palabras he utilizado los listados que he encontrado en una web²⁰ que ha clasificado casi todas las palabras en listados, añadiendo las etiquetas EAGLE a cada una de ellas.

He almacenado cada uno de estos listados, con sus etiquetas EAGLE en ficheros con formato csv., separados por un punto y coma.

²⁰ http://www.solosequenosenada.com/nexus/index.php?title=Ortografia_esp%C3%B1ola_EAGLE

7.2. Web Scraping

Web scraping es una técnica utilizada mediante programas de software para extraer información de sitios web. Usualmente, estos programas simulan la navegación de un humano.

Es un campo con desarrollos activos, compartiendo un propósito en común con la visión de la Web semántica. Existen distintos niveles de automatización que las existentes tecnologías de Web Scraping pueden brindar.

- Algoritmos de minería de datos: muchos sitios webs tienen grandes colecciones de páginas generadas dinámicamente a partir de una base de datos. Datos de la misma categoría aparecen usualmente en páginas similares mediante un script o una plantilla. En la minería de datos, un programa detecta estas plantillas en un contexto específico y extrae su contenido.
- Reconocimiento de información semántica: las páginas que son analizadas podrían incluir metadatos o cierta información semántica como anotaciones o comentarios, los cuales pueden ser usados comúnmente. Si estas anotaciones están en las mismas páginas, como sucede con los microformatos, estas podrían ser de utilidad cuando parseamos el DOM del documento. En otro caso, las anotaciones, organizadas en una capa semántica, son almacenadas y manejadas de forma separada desde otras páginas, por lo que los scrapers pueden recuperar estos esquemas y las instrucciones desde esta capa antes de analizar los documentos.

7.2.1. El web scraping como sustituto de una API

Una API²¹ (Application Programming Interface) es una herramienta que nos permite intercambiar datos entre varias webs. Si la web de donde queremos obtener los datos no dispone de una API, podemos utilizar un web scrapper para obtener los datos necesitados, siempre comprobando que esta técnica esté permitida en el sitio web.

²¹ https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones

7.3. Desarrollo de los métodos (Back-End)

7.3.1. Separar frases

Para poder realizar de una manera más eficiente la simplificación del texto, separo el texto introducido por el usuario en frases más cortas, basándome en los puntos finales de las frases, una vez detecto un punto, fraccio la frase y la almaceno en un objeto tipo “frases”, trabajando sobre las frases sueltas, una a una.

```
public class Frases
{
    public string Frase {get; set;}
    public string FraseLimpia {get; set;}
    public bool IsPasiva {get; set;}
    public List<Palabra> Palabras {get; set;}
}
```

La frase original se almacena en el campo “Frase”, una vez almacenada, me dispongo a realizar los cambios necesarios para obtener la simplificación (si fuese necesaria) de la misma.

7.3.2. Limpiar frase

Una vez tenemos la frase introducida en el objeto, paso a limpiar la frase, para ello, busco en la frase, a través de varias expresiones regulares, si existe texto entre paréntesis, guiones o símbolos “<” “>”.

Este paso sirve para eliminar contenido que para la comprensión del texto no es relevante y puede hacer que la lectura y comprensión de la frase, sea más compleja.

Una vez se ha limpiado la frase, introducimos el resultado en el campo “FraseLimpia” del objeto “Frases” que hemos creado anteriormente.

En estos momentos, tenemos la frase original introducida por el usuario y la frase sin paréntesis u otros signos que cumplan el mismo cometido que los paréntesis.

7.3.3. Quitar conectores de ejemplificación

Una vez tenemos la frase limpia de paréntesis y demás, pasamos a buscar conectores de ejemplificación, que, aunque en ocasiones pueden ayudar a la comprensión del texto, en su mayoría, son frases que se pueden eliminar del texto, haciendo que este sea más corto y a su vez, más sencillo de leer.

Para ello, he montado un archivo en formato csv donde están todos los conectores de ejemplificación, por lo que busco en la frase si existe alguno de ellos y si encuentro en la frase alguno de estos conectores, los elimino de la frase, volviendo a almacenar en el campo de “FraseLimpia” la frase sin los conectores de ejemplificación.

7.3.4. Obtención de las palabras de la frase

Una vez tenemos limpia la frase, la recorremos, separando y añadiendo a un objeto de tipo “Palabra” cada una de las palabras de la frase.

```
public class Palabra
{
    public int Id {get; set;}
    public string Nombre {get; set;}
    public Adverbios Adverbio {get; set;}
    public Adjetivos Adjetivo {get; set;}
    public Articulos Articulo {get; set;}
    public Conjunciones Conjuncion {get; set;}
    public Preposiciones Preposicion {get; set;}
    public Pronombres Pronombre {get; set;}
    public Sustantivos Sustantivo {get; set;}
    public Verbos Verbo {get; set;}
}
```

En este objeto, tendremos la palabra y el tipo de palabra de la que se trata, las clases que componen la clase palabra tienen los datos específicos de las etiquetas EAGLE para su tipo.

Como ejemplo, seguiremos mostrando el caso para los adjetivos, teniendo en cuenta que esto se repite para cada una de las clases que componen la clase “Palabra”.

```
public class Adjetivos
{
    public EnumCategoriaPalabras Categoria {get; set;}
    public EnumTipoPalabras Tipo {get; set;}
    public string Grado {get; set;}
    public EnumGeneroPalabra Genero {get; set;}
    public EnumNumeroPalabras Numero {get; set;}
    public bool Participio {get; set;}
}
```

Cada palabra es buscada en los archivos csv de nuestro corpus de tipos de palabras, se busca en todos los archivos ya que una misma palabra puede existir en varias listas del corpus, una vez la encuentra, añade al objeto correspondiente los valores dependiendo de la etiqueta EAGLE que tenga dicha palabra.

7.3.5. Comprobar si la frase está en pasiva

Para distinguir si una frase está en pasiva o no, se ha de seguir la regla que nos marca la RAE²² (Real Academia Española) en su punto 38.

“Tienen valor pasivo o están en forma pasiva las oraciones o construcciones que tienen un sujeto paciente [...] La pasiva se construye mediante una perífrasis formada por el auxiliar ser y el participio del verbo principal, el cual concuerda con el sujeto.”

Por ello, comprobamos si existe en la frase un verbo “ser” en cualquiera de sus conjugaciones, para ello, nos valemos del corpus de verbos que tenemos almacenado, si encontramos alguna de sus conjugaciones en la frase, comprobamos que la siguiente palabra, es un verbo y se encuentre conjugado en participio, si cumple esta estructura, podremos asegurar un alto porcentaje de las veces de que se trata de una frase en voz pasiva.

²² <https://www.rae.es/diccionario-panhispanico-de-dudas/terminos-linguisticos>

Una vez sabemos que la frase se encuentra en voz pasiva, tenemos que obtener el verbo principal, que está en participio, en la conjugación correcta, para ello, hacemos uso de WordReference²³. Se hace una llamada a la web y a través de Web Scraping²⁴ obtenemos infinitivo del verbo principal.

Una vez tenemos el verbo principal en su forma infinitiva, utilizaremos las etiquetas EAGLE del verbo “ser” para obtener el tiempo, número y género del verbo principal, es decir, si el verbo ser está conjugado en pasado y en singular, obtendremos con la misma técnica de Web Scraping el verbo principal en pasado y singular.

Solo falta separar el sujeto y el complemento, cambiarles el orden y añadir el verbo principal en su conjugación correcta entre ambos, para ello, se almacena en una variable string todas las palabras hasta que nos encontramos con el verbo “ser” y en otra variable, también de tipo string, todas las palabras después del verbo principal.

Una vez tenemos todas las partes principales de la frase, podemos juntarlas en el orden correcto (sujeto + verbo + complemento) para formar una frase en voz activa según la definición que podemos obtener de la RAE para la voz activa.

“Tienen valor activo o están en forma activa las oraciones o construcciones cuyo sujeto designa la entidad que desempeña el papel más activo en la acción, proceso o situación expresados por el verbo. [...] En un sentido amplio, se consideran activas todas las oraciones que no están en forma pasiva.”

²³ <https://www.wordreference.com/conj/EsVerbs.aspx>

²⁴ https://es.wikipedia.org/wiki/Web_scraping

7.3.6. Obtener sinónimos

Para la obtención de los sinónimos, hago uso de la técnica de Web Scraping ya explicada, para ello, hago una petición a la página web de WordReference a la sección de sinónimos²⁵ a la que se le añade al final de la dirección la palabra de la cuál queremos saber sus sinónimos, una vez encontrados, los añado a una lista y la muestro.

7.3.7. Obtener definiciones

Para la obtención de las definiciones, se utiliza de nuevo, la técnica de Web Scraping hacia la página de WordReference, a la sección de definiciones²⁶ añadiendo al final de la dirección la palabra de la que queremos saber su definición, una vez encontramos la definición, la añadimos a una lista y se muestra.

7.3.8. Obtener imágenes palabra

Para obtener la imagen más significativa de la palabra buscada, hago uso de las imágenes de Google, mediante el uso de Web Scraping, mostrando la primera imagen que aparece en Google imágenes en primera posición.

²⁵ <https://www.wordreference.com/sinonimos/sin%C3%B3nimo>

²⁶ <https://www.wordreference.com/definicion/>

7.4. Desarrollo de la interfaz (Front-End)

7.4.1. Pantalla principal

En el siguiente apartado voy a mostrar lo que sería un ejemplo de ejecución desde la aplicación, mostrando las diferentes posibilidades que se puede encontrar un usuario.

Al entrar en la web, la aplicación muestra al usuario todas las opciones de las que dispone.



Figura 11 Interfaz EasyText

7.4.2. Menú lateral

A la derecha, podemos observar una barra de menú, donde aparece, el nombre de la aplicación, un botón de vuelta a la página principal y tres botones que sirven para aumentar, disminuir y volver al tamaño original el texto de toda la web.

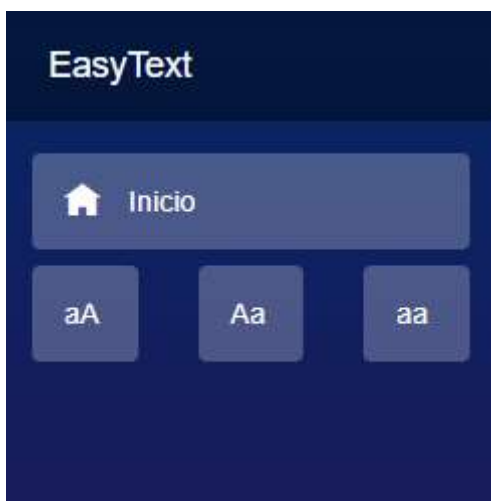


Figura 12 Menú lateral

Los botones situados debajo del botón de inicio, como se ha comentado anteriormente, sirven para aumentar, disminuir y volver al tamaño original del texto. Esto se puede ver más claramente, si dejamos unos segundos el ratón encima de cada uno de ellos.



Figura 13 Hint botón aumentar texto

7.4.3. Búsqueda de sinónimos


Para la búsqueda de sinónimos tenemos un pequeño input donde podemos introducir la palabra de la cuál queremos saber sus sinónimos junto con un botón para iniciar la búsqueda.



The screenshot shows a web interface for finding synonyms. At the top, there is a blue horizontal bar with the word "Sinónimos" in white text. Below this bar, there is a white search area. On the left side of this area is a text input field with the placeholder text "Sinonimo". To the right of the input field is a blue button with the text "Buscar" in white.

Figura 14 Búsqueda de sinónimos

Una vez introducimos una palabra y presionamos el botón de buscar, aparecerán los sinónimos en una lista justo debajo de la sección de búsqueda de sinónimos.



The screenshot shows the same web interface as Figure 14, but now displaying search results. The blue header "Sinónimos" remains at the top. Below it, the search input field now contains the word "Formulario". The "Buscar" button is highlighted with a blue border. Below the search area, there is a list of four synonyms, each on a separate line with a right-pointing arrow:

manual	▶
prontuario	▶
impreso	▶
recetario	▶

Figura 15 Resultado búsqueda sinónimos

7.4.4. Búsqueda de definiciones

Al igual que para la búsqueda de sinónimos, las definiciones están planteadas de la misma manera, introducimos la palabra de la que queremos saber su definición en un input preparado para ello y presionamos el botón de buscar.

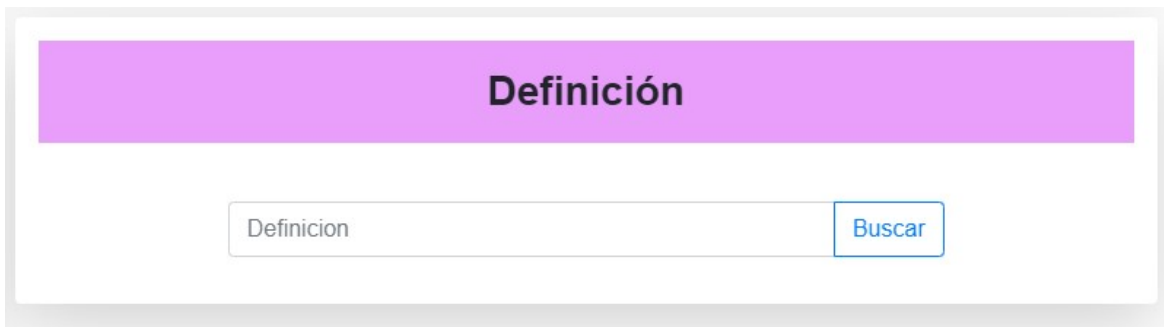


Figura 16 Búsqueda de definiciones

Una vez introducimos la palabra de la que queremos ver su definición, se mostrará justo debajo de la sección de búsqueda de definiciones la definición de la palabra buscada, junto con una pequeña imagen descriptiva que nos puede servir de ayuda para interpretar mejor la palabra.

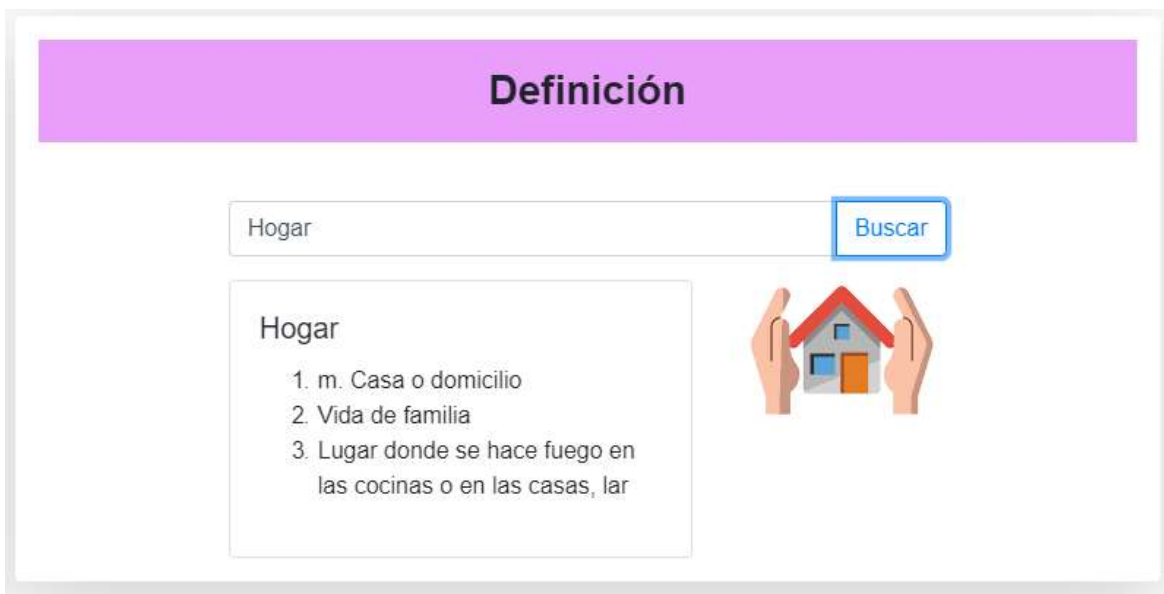


Figura 17 Resultado de búsqueda de definiciones

Por otra parte, tenemos la posibilidad de ver la definición de los sinónimos que hemos buscado anteriormente, para ello, solo deberemos presionar en la flecha (▶) que aparece al lado de cada sinónimo.

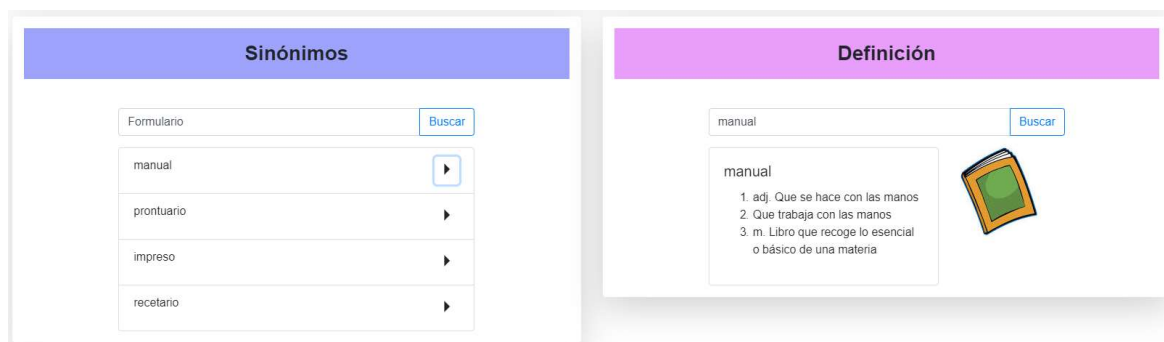
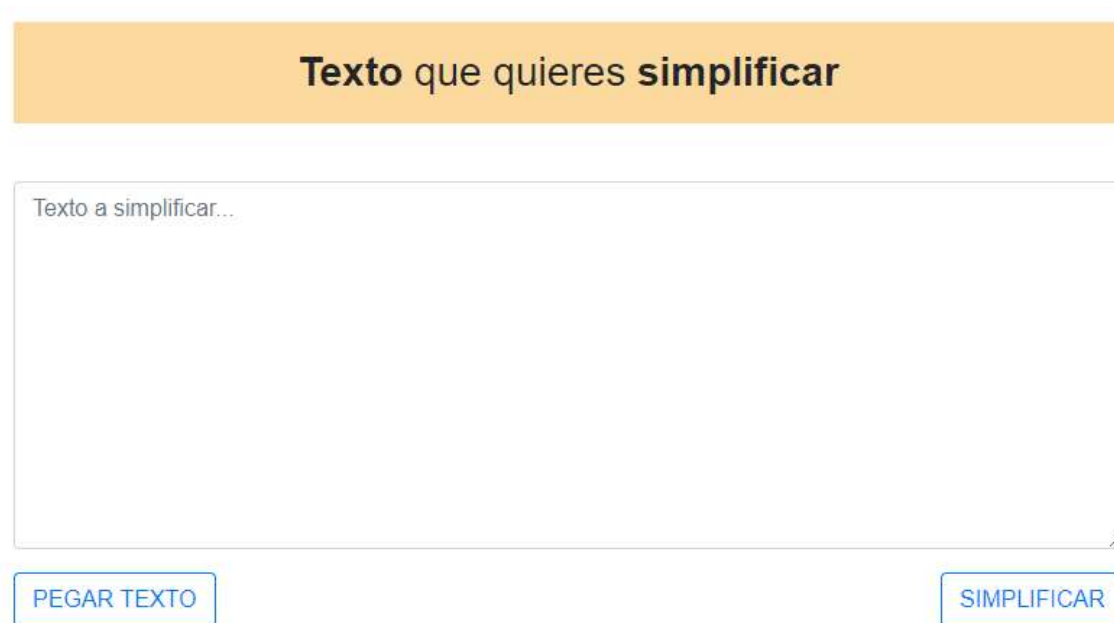


Figura 18 Definición de los sinónimos

7.4.5. Simplificar un texto

En este apartado es donde introduciremos el texto que queremos simplificar, bien escribiéndolo o bien si ya lo tenemos copiado en el portapapeles, pegándolo con el uso del botón “Pegar texto”.



Texto que quieres simplificar

Texto a simplificar...

PEGAR TEXTO SIMPLIFICAR

Figura 19 Sección para simplificar textos

Una vez se introduce el texto, para simplificarlo, simplemente debemos presionar el botón de “Simplificar” apareciendo el texto simplificado, justo al lado del texto original, en la sección adecuada para ello.

Texto simplificado

Figura 20 Sección del texto simplificado

Una vez introducimos un texto y presionamos el botón de simplificar, se mostrará el resultado del texto simplificado.

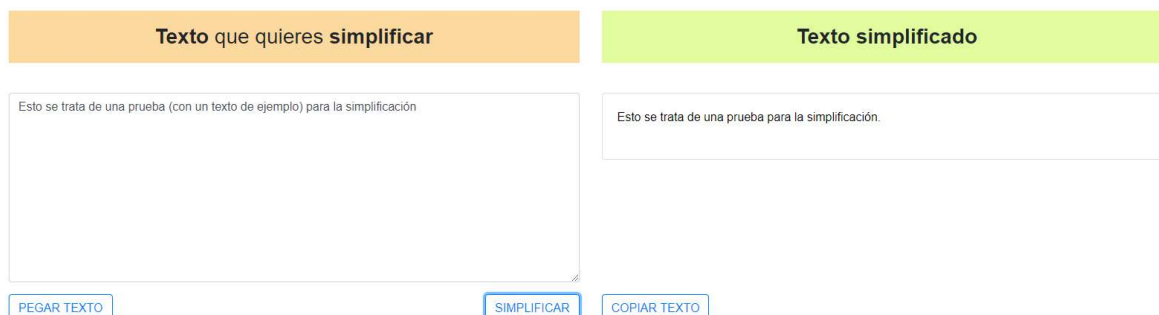


Figura 21 Resultado de la simplificación

Una vez hemos realizado la simplificación del texto, aparece justo debajo del texto simplificado un botón que copia el texto simplificado al portapapeles, para poder añadirlo a un documento de una manera mucho más rápida.

Otra de las opciones que de las que dispone es la de mostrar la definición y sinónimos de cualquier palabra que el texto simplificado nos devuelve.



Figura 22 Sinónimos/definiciones al pulsar en texto simplificado

8. Evaluación y resultados

Para poder evaluar los métodos desarrollados y poder comprobar que su funcionalidad es la correcta debo tener un conjunto de pruebas (textos) que cubran el máximo porcentaje posible todos ellos.

8.1. Quitar texto entre paréntesis y otros símbolos

TEXTO ORIGINAL	TEXTO OBTENIDO	RESULTADO
Esto es un texto de prueba que funciona (espero) correctamente	Esto es un texto de prueba que funciona correctamente.	OK
Esto es un texto de prueba que funciona -espero- correctamente	Esto es un texto de prueba que funciona correctamente.	OK
Esto es un texto de prueba que <<espero>> funciona correctamente	Esto es un texto de prueba que funciona correctamente.	OK
Esto es un texto de prueba que —espero— funciona correctamente	Esto es un texto de prueba que funciona correctamente.	OK
Esto es un texto de prueba que <espero> funciona correctamente	Esto es un texto de prueba que funciona correctamente.	OK
Prueba de texto con un guión - < << > > para comprobar su funcionamiento	Prueba de texto con un guión - para comprobar su funcionamiento.	OK
Hay dos grandes corrientes de autoconciencia, el <<yo>> (que nos remite al presente) y el <<mí>> (que elabora relatos sobre nuestro pasado y nuestro futuro).	Hay dos grandes corrientes de autoconciencia, el y el.	KO
Los artistas del circo se esfuerzan en lograr lo que ellos llaman una <<práctica perfecta>>	Los artistas del circo se esfuerzan en lograr lo que ellos llaman una.	KO
Las <<sensaciones viscerales>> son mensajes procedentes de la ínsula	Las son mensajes procedentes de la ínsula	KO

Tabla 4 Casos de pruebas para eliminar paréntesis y otros símbolos

Se han probado otros símbolos, además de los paréntesis como los guiones simples (-), mayor y menor (<>), mayor y menor doble (<<>>) o guiones dobles (—).

Durante las pruebas se ha detectado que el uso de los mayores y menores dobles (<<>>) cuando son usados para hacer hincapié en ciertas palabras no se pueden eliminar, ya que harían que la frase perdiera totalmente el sentido.

Después de realizar las pruebas y debido a que el uso que se le da a estos símbolos (<<>>) son mayoritariamente para el remarcado de palabras clave en el texto, se ha tomado la decisión de no eliminarlos del texto ya que su eliminación hace perder completamente el sentido a la frase.

8.2. Eliminar conectores de ejemplificación

TEXTO ORIGINAL	TEXTO OBTENIDO	RESULTADO
Estas serán las únicas modificaciones de un espectáculo que mantiene su esencia, explicó el primer edil, con pequeños cambios con base en la experiencia de los pirotécnicos, para resumir, el ensayo y error que se realiza cada año	Estas serán las únicas modificaciones de un espectáculo que mantiene su esencia, explicó el primer edil, con pequeños cambios con base en la experiencia de los pirotécnicos.	OK
La lengua española es un lenguaje rico en conjugaciones verbales, por ejemplo, tenemos las conjugaciones verbales en pasado	La lengua española es un lenguaje rico en conjugaciones verbales.	OK
Las figuras geométricas son aquellas formas básicas utilizadas en la geometría, por ejemplo: el triángulo, el rectángulo, el cuadrado.	Las figuras geométricas son aquellas formas básicas utilizadas en la geometría.	OK
Queremos expresar nuestro apoyo a la propuesta escolar expuesta por nuestro director. En suma, estamos plenamente de acuerdo con las modificaciones planteadas.	Queremos expresar nuestro apoyo a la propuesta escolar expuesta por nuestro director.	OK
Los avances de la ciencia son notables en los últimos 100 años y así como hemos eliminado la viruela, ahora podremos eliminar las enfermedades del siglo XXI.	Los avances de la ciencia son notables en los últimos 100 años y.	KO

Tabla 5 Casos de prueba de conectores

Se han probado varios casos de conectores, tanto de ejemplificación como de resumen, en casi todos los casos, las pruebas han sido satisfactorias, salvo en los casos en los que antes del conector no existe una parada sintáctica, como puede ser un punto o una coma, en esos casos, aunque es cierto que elimina correctamente el conector, la frase no queda bien formada del todo, como es el caso del último caso de prueba para este método.

8.3. Cambio de frases voz pasiva a activa

TEXTO ORIGINAL	TEXTO OBTENIDO	RESULTADO
El asunto fue delimitado por todos los diputados	Todos los diputados delimitaron el asunto	OK
Las decisiones fueron tomadas	Tomaron las decisiones	OK
Las decisiones fueron tomadas por mi	Mi tomaron las decisiones	KO
El camino fue recorrido por los animales	Los animales recorrieron el camino	OK
La comida fue cocinada por los chefs	Los chefs cocinaron la comida	OK

Tabla 6 Caso de pruebas para cambio de voz pasiva a activa

Para este método, aunque hace correctamente el paso de la voz pasiva a la voz activa, se han detectado pequeños errores al pasar el sujeto pasivo, que es quien recibe la acción, al sujeto activo, quien realiza la acción.

Para el resto de los casos, podemos comprobar que realiza correctamente el paso de la voz pasiva a voz activa.

9. Conclusiones y trabajo futuro

Como ya hemos visto, el acceso a la información es un derecho fundamental para todas las personas y por ello, se debe garantizar este derecho para todos los colectivos, por ello, la aplicación diseñada y desarrollada, aunque con mucho por mejorar y añadir, puede hacer que las personas que por cualquier motivo ya mencionado no puedan acceder correctamente a la información, estén algo más cerca de conseguirlo.

Gracias a las pruebas realizadas, podemos ver que los métodos hasta ahora desarrollados funcionan correctamente, haciendo que se cumplan tanto el objetivo principal como los objetivos específicos. Por lo que podemos decir que el resultado ha sido satisfactorio.

Las mejoras que se pueden realizar a la aplicación son muchas y de distintos tipos, podemos seguir mejorando la simplificación del texto, separando de una manera más eficiente las frases complejas, mejorando los métodos ya existentes para obtener un mejor resultado.

Por otro lado, las mejoras pueden dirigirse hacia la accesibilidad de la web a otros colectivos con dificultades visuales, como por ejemplo, a las personas con daltonismo, añadiendo la funcionalidad de cambiar el color del fondo de la aplicación para que este colectivo, pueda ver la información sin ningún tipo de problemas.

Bibliografía

- Actas del Sexto Congreso de Hispanistas de Asia*. Anula. 2008. 2008. Tipos de textos, complejidad lingüística y facilitación lectora. págs. Páginas 45-61.
- Alúísio, Sandra M., y otros.
- Candido, Arnaldo, y otros. *Supporting the Adaptation of Texts for Poor Literacy Readers*.
- Carroll, John, y otros. 1998. *Practical Simplification of English Newspaper Text to Assist Aphasic Readers*. 1998.
- Freyhoff, G., y otros. 1998. *European Guidelines for the Production of Easy-to-Read Information for People with Learning Disability*. ISLMH : s.n., 1998. Informe Técnico.
- Freyhoff, Geert, y otros. 1998. El Camino Más Fácil - Directrices Europeas para Generar Información de Fácil Lectura.
<http://www.lecturafacil.net/media/resources/ILSMHcastellà.pdf>. [En línea] 1998.
- Quinlan, Philip. 1992. *The Oxford Psycholinguistic Database*. Oxford : Oxford University Press, 1992.
- Rodríguez Diéguez, José Luis, Moro Berihuete, Pilar y Cabero Pérez, María Visitación. 1993. *Ecuaciones de predicción de lecturabilidad. Informe técnico*. UNED : e-spacio, 1993.