



Facultad de Ciencias

Aplicación web para traducir documentos a través de un flujo de trabajo

(Web application to translate documents through a workflow)

Trabajo de fin de grado
para acceder al

GRADO EN INGENIERÍA INFORMÁTICA

Autor: Mónica Casanova Mediavilla

Director: Cristina Tirnauca

Co-director: Miguel Cortés Tazón

Junio de 2020

Agradecimientos

A mi familia, por haberme acompañado y apoyado toda mi vida, por haberme educado y ayudado a convertirme en la persona que soy hoy en día y por haberme alentado en todo lo que he hecho.

A mis amigos y muchos de ellos compañeros de carrera, en los que incluyo a mi pareja, por levantarme el ánimo en los momentos más difíciles y apoyarnos mutuamente siempre.

A mi directora de proyecto y profesora durante la carrera Cristina Tirnauca, por ayudarme en este proyecto hasta el último momento.

A mi co-director Miguel Cortés y compañeros de trabajo en Incentro, por ayudarme siempre que lo he necesitado durante el desarrollo de la aplicación.

A todos los profesores con los que he coincidido en la carrera, por haberme aportado conocimientos para comenzar mi vida laboral en la informática.

Resumen

palabras clave: Aplicación web, Traducción, Alfresco

La traducción está presente en muchas empresas que tienen oficinas repartidas por el mundo o que trabajan con otras con las que no comparten su lengua materna. Por ello es importante tener buenas herramientas que les permitan traducir documentos en todos los idiomas que necesiten.

Esas mismas empresas suelen tener una gran cantidad de información que necesitan tener localizada y bien organizada. Una solución para este problema viene dada por los gestores de contenido que proporcionan el almacenamiento, organización y clasificación de la información, entre otras cosas.

Para realizar el proyecto se va a usar Alfresco Content Services, un potente gestor de contenido en auge, que se usa con frecuencia tanto en el sector público como en el privado. También se utilizarán Alfresco Process Services, para la creación de un Modelo de Proceso de Negocio a través del cual se puedan traducir y revisar documentos, y Alfresco Application Development Framework para el desarrollo de una aplicación web que facilite la interacción con los documentos, su traducción y revisión. Además, la traducción del contenido de los documentos será llevada a cabo por Google Translation API.

(Web application to translate documents through a workflow)

Abstract

keywords: Web application, Translation, Alfresco

Many companies that have their headquarters distributed all over the world or that work with others in which the mother tongue of their employees is different are constantly making use of translations. Thus, it is very important for them to have suitable tools to perform translations into all the languages they need.

Those same companies often have a great amount of information that needs to be easily locatable and well organize. A solution for this problem is given by content management systems that provide storage, organization and sorting of the information, among other things.

For this project we will use Alfresco Content Services, a powerful booming content management system. For the creation of a Business Process Model, Alfresco Process Services will be used. That process will include the translation and review of documents. Also, Alfresco Application Development Framework will be used for the development of the web application that supplies the user interaction with the documents, their translation and review. Furthermore, the translation of the document will be carried out by Google Translation API.

Índice

1. Introducción	1
1.1. Contexto	1
1.2. Objetivos	1
2. Tecnologías, herramientas y lenguajes utilizados	3
2.1. Tecnologías y Herramientas	3
2.2. Lenguajes	4
3. Metodología y planificación	7
3.1. Metodología	7
3.2. Planificación	7
4. Presentación del problema y especificación de requisitos	11
4.1. Presentación del problema	11
4.2. Análisis de requisitos	11
4.2.1. Actores del sistema	11
4.2.2. Requisitos funcionales	11
4.2.3. Requisitos no funcionales	12
5. Especificación de casos de uso	13
5.1. Diagramas de casos de uso	13
5.1.1. Diagrama para el rol Administrador	13
5.1.2. Diagrama para el rol Colaborador	14
5.1.3. Diagrama para el rol Editor	14
5.2. Plantillas de casos de uso	15
6. Creación de los entornos de trabajo	19
7. Diseño e implementación	21
7.1. Diseño arquitectónico	21
7.2. Capa de presentación	22
7.2.1. Vista de la página de inicio de sesión	22
7.2.2. Vista de la página de contenido	22
7.2.3. Vista de la página de tareas	24
7.3. Capa de negocio	24
7.3.1. Traducción de documentos	25
7.3.2. Proceso	28
7.4. Capa de datos	29
8. Pruebas y resultados	31
8.1. Pruebas unitarias	31
8.2. Pruebas de integración	32
8.3. Pruebas de sistema	32

8.3.1. Pruebas de rendimiento	32
8.3.2. Pruebas de usabilidad	33
8.4. Pruebas de aceptación	33
9. Conclusiones y trabajos futuros	35
10. Bibliografía	37

1 Introducción

1.1. Contexto

A día de hoy la traducción es necesaria en muchas empresas ya que disponen de oficinas en distintos puntos del mundo o trabajan en el mercado internacional. Aunque el uso del inglés esté bastante extendido y sea el idioma elegido para las comunicaciones internacionales, la comprensión y facilidad para expresarse en la lengua materna de cada uno siempre será mayor. Por ello es importante para una empresa tener herramientas que faciliten la traducción y ayuden a sus empleados a desarrollar su trabajo sin impedimentos.

Por otro lado, debido a la gran cantidad de información que generan las empresas, estas necesitan una herramienta con la que poder tener localizada y controlada dicha información. Una de las posibles herramientas a utilizar es un *Enterprise Content Management* (ECM). Los ECM facilitan el almacenamiento, seguridad, control de versiones, organización y clasificación de la información, lo que aporta una reducción de costes al poder prescindir de los documentos físicos.

Uno de los grandes gestores del mercado actualmente es Alfresco. Este software destaca sobre los demás por ser de código abierto, lo que permite su compatibilidad con otros entornos o aplicaciones. También permite realizar búsquedas a través de los metadatos del contenido y así obtener resultados en base a detalles más precisos de los archivos y no solo por su nombre o fecha de creación. Además, posibilita la creación y gestión de flujos de trabajo que permiten automatizar procesos empresariales, lo cual es una forma de ahorrar tiempo en procedimientos en los que están involucradas varias personas y que tradicionalmente se realizan a través de correos electrónicos.

Un ejemplo de utilidad de la combinación de un ECM y la traducción en una consultoría sería el traspaso de documentación sobre un proyecto entre dos oficinas en las que el idioma principal es diferente. En el caso de Incentro EIM Services, la oficina de Santander realiza el mantenimiento del software desarrollado en diferentes proyectos de Holanda. En la mayoría de los proyectos, la documentación está escrita en holandés, por lo que es necesaria su traducción para que los encargados del mantenimiento de dicho software puedan comprenderlo al máximo.

1.2. Objetivos

El objetivo principal de este trabajo de fin de grado es desarrollar una aplicación web en la que el usuario pueda almacenar sus documentos y que le permita traducir documentos de texto plano y word. Esta aplicación facilitará agilizar y optimizar el proceso de traducción de documentos. En ella se almacenarán tanto los documentos que se quieren traducir como los ya traducidos. A continuación se enumeran los sub-objetivos del proyecto:

- Proporcionar una aplicación web en la que almacenar contenido variado como documentos words, pdf o video.
- Permitir la traducción de los documentos de texto plano y word a través de una acción.
- Ofrecer una forma de revisión de los documentos una vez traducidos en la que poder realizar cambios para la debida corrección de la traducción.

- Facilitar la manipulación del contenido a través de acciones como visualizar, descargar, eliminar, mover y copiar.

2 Tecnologías, herramientas y lenguajes utilizados

En este apartado se explican las tecnologías y herramientas utilizadas, así como los lenguajes de programación empleados para el desarrollo de la aplicación.

2.1. Tecnologías y Herramientas

A continuación se da una breve explicación de todas las tecnologías y herramientas utilizados a lo largo del desarrollo de este proyecto.

- Alfresco Content Services (ACS)

Es un gestor de contenido empresarial (*Enterprise Content Management*, ECM) que ayuda a las empresas a recopilar, administrar, almacenar, preservar y poner a disposición de los usuarios los innumerables fragmentos de información electrónicos y físicos que proliferan en una empresa, desde mensajes de correo electrónico hasta vídeos digitales, pasando por archivos en papel y el cada vez mayor número de documentos de oficina [1].

- Alfresco Process Services (APS)

Es una solución de gestión de procesos empresariales (*Business Process Model*, BPM) orientada a gente de negocios y desarrolladores. En su núcleo se encuentra un motor de procesos empresariales de código abierto y alto rendimiento basado en Activiti con flexibilidad y escalabilidad para manejar una amplia variedad de procesos críticos. Activiti [2] es un motor de BPM de código abierto y centrado en Java que admite las necesidades de automatización de procesos del mundo real. Alfresco Process Services ofrece un potente conjunto de herramientas para usuarios finales y se integra en una amplia variedad de sistemas empresariales, incluidos Alfresco Content Services, Box y Google Drive [3].

- Alfresco Application Development Framework (ADF)

Es un framework moderno basado en tecnologías JavaScript, para crear de forma rápida atractivas aplicaciones web utilizando el potencial de Alfresco Digital Business Platform. Proporciona un amplio conjunto de componentes y servicios reutilizables para una interfaz de usuario basada en Angular, herramientas de línea de comandos y API (*Application Program Interface*) de JavaScript que completan la combinación de los servicios de gobierno, procesos y contenido de Alfresco. Permite la creación de aplicaciones web interactivas [4]. Una API es una interfaz de programación de aplicaciones que expone un conjunto de datos y funciones para facilitar las interacciones entre programas [5].

- Angular

Es un framework de diseño de aplicaciones y una plataforma de desarrollo para crear aplicaciones eficientes y sofisticadas de una sola página [6].

- **Google Translation API**

Es un servicio proporcionado por Google que permite la detección del idioma de un texto y su posterior traducción a otro idioma. El texto de entrada puede ser HTML o texto sin formato. No traduce ninguna etiqueta HTML en la entrada, solo el texto que aparece entre las etiquetas [7].
- **Oracle VM VirtualBox**

Es una aplicación de virtualización multiplataforma. Permite ejecutar diferentes sistemas operativos, dentro de múltiples máquinas virtuales, al mismo tiempo [8].
- **MobaXterm**

Es un escritorio remoto que proporciona todas las herramientas de red remota (SSH, RDP, X11, SFTP, FTP, Telnet, Rlogin, ...) al escritorio de Windows [9].
- **IntelliJ IDEA**

Es un entorno de desarrollo integrado (IDE) multiplataforma para lenguajes JVM [10].
- **Visual Studio Code**

Es un editor de código fuente ligero pero potente que se ejecuta en modo escritorio y está disponible para Windows, macOS y Linux [11].
- **Git**

Es un sistema de control de versiones distribuido, gratuito y de código abierto diseñado para manejar desde proyectos pequeños a grandes con rapidez y eficiencia [12].
- **SourceTree**

Es un cliente gratuito de Git. Proporciona una interfaz gráfica que facilita la interacción con Git [13].
- **Postman**

Es una plataforma de colaboración para el desarrollo de APIs. Las características de Postman simplifican cada paso de la creación de una API y agilizan la colaboración para poder crear mejores APIs más rápido [14].
- **Node.js**

Es un entorno de ejecución de JavaScript orientado a eventos asíncronos, está diseñado para crear aplicaciones *network* escalables [15].
- **Yeoman**

Es una herramienta que permite generar proyectos en cualquier lenguaje. Es básicamente un *plug-in* que se ejecuta con el comando “yo” para montar proyectos completos [16].
- **Maven**

Apache Maven es una herramienta de gestión y comprensión de proyectos de software. Basado en el concepto de un modelo de objeto de proyecto (POM), Maven puede administrar la construcción, los informes y la documentación de un proyecto a partir de una información central [17].

2.2. Lenguajes

En esta sección se explica brevemente cada uno de los lenguajes empleados en la capa de *backend* como en la de *frontend*.

- Java

Es la base para prácticamente todos los tipos de aplicaciones de red, además del estándar global para desarrollar y distribuir aplicaciones móviles y embebidas, juegos, contenido basado en web y software de empresa [18].

- HTML

Es el lenguaje de marcado estándar para crear páginas web. Describe la estructura de una página web [19].

- TypeScript

Es un lenguaje de programación de código abierto desarrollado por Microsoft, el cual cuenta con herramientas de programación orientada a objetos [20]. Es un superconjunto de JavaScript que se compila en JavaScript simple [21].

- CSS

Es un lenguaje que describe el estilo de un documento HTML. Describe cómo se deben mostrar los elementos HTML [22].

3 Metodología y planificación

En este apartado se explica la metodología y la organización seguida durante el desarrollo del proyecto.

3.1. Metodología

Para la realización de este proyecto se ha utilizado el modelo de desarrollo incremental [23]. Este tipo de desarrollo consiste en la realización de diferentes versiones del sistema hasta conseguir la versión final. Cada versión engloba fases de especificación, de desarrollo y de validación. Una vez completada la versión, esta se expone al usuario para recoger el *feedback*. Seguir esta metodología en un proyecto permite que el usuario obtenga una mejor comprensión del sistema además de reducir el coste de adaptar los cambios que solicite el usuario. Se ha podido utilizar esta metodología porque el contacto con el co-director del proyecto ha sido frecuente y se podían realizar las reuniones necesarias en cada incremento.

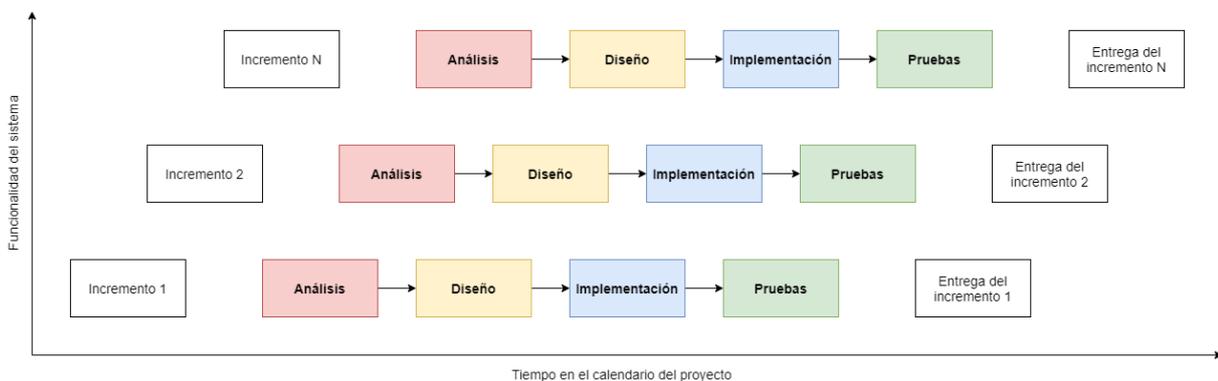


Figura 1: Funcionamiento del modelo de desarrollo incremental.

3.2. Planificación

La realización de este proyecto se ha llevado a cabo trabajando en la empresa Incentro EIM Services, comenzando en octubre dicho trabajo. La ejecución de este no comenzó hasta febrero, ya que algunos de los conocimientos necesarios no fueron adquiridos hasta ese momento. Aún así, muchos de ellos se han adquirido a medida que se ha ido avanzando ya que en la empresa el trabajo que se realiza es de mantenimiento y este ha sido un desarrollo desde cero.

Como se ha explicado en la anterior sección, se ha empleado un modelo de desarrollo incremental en el que la división del proyecto ha sido la siguiente: en una primera fase, se realizó un análisis de requisitos y diseño del sistema en términos generales, se investigaron y probaron las tecnologías que se utilizarían, y además se instalaron y configuraron las máquinas virtuales para la ejecución de ACS y APS y se configuró el proyecto de ADF.

Posteriormente, el desarrollo e implementación se ha dividido en 4 incrementos. El primer incremento corresponde al desarrollo de un web script para la traducción de documentos en la parte de ACS, el segundo con el desarrollo del flujo de trabajo en la parte de APS, el tercero con el desarrollo del *frontend* que corresponde con el inicio de sesión y la lista de contenido en la parte de ADF y el cuarto con el desarrollo del *frontend* que corresponde a la lista de tareas y la vista de estas.

A continuación se especifica con un poco más de detalle el trabajo desarrollado en cada uno de los incrementos:

- Incremento 1: En este primer incremento se desarrolla un web script para la traducción de documentos de texto plano (documentos con extensión .txt) y word (documentos con extensión .docx). Para la traducción de estos dos tipos de documentos se utiliza la Google Translation API que permite traducir texto plano y HTML (traduce el contenido entre las etiquetas HTML). Por tanto, el documento txt se puede traducir directamente mientras que para el word hay que convertirlo a HTML primero. Todo esto se implementa en el código que se despliega en la máquina virtual en la que está instalado ACS.
- Incremento 2: El objetivo de este incremento es crear un proceso (BPM) con el que traducir y posteriormente revisar el documento indicado. Está compuesto por un formulario en el evento de inicio del proceso y dos tareas. En la primera se llama al web script creado en el anterior incremento. En la segunda se crea un formulario en el que descargar el documento traducido y subir una nueva versión del mismo; estas dos acciones del formulario solo estarán disponibles en la aplicación en ADF (*frontend*).
- Incremento 3: En este incremento se comienza a desarrollar la interfaz de la aplicación web. Por una parte se personaliza la vista del inicio de sesión y se realizan las acciones necesarias para poder iniciar sesión contra ACS y APS desde la aplicación web. Por otra parte, se crea la lista de sitios y contenido de los sitios, con las correspondientes acciones sobre el diferente tipo de contenido.
- Incremento 4: En este último incremento se finaliza el desarrollo de la interfaz de la aplicación web. Se crea la vista de las tareas divididas en dos categorías: asignadas al usuario y reclamables por el usuario. Además se completa el formulario creado en el Incremento 2 para revisar el documento, añadiendo los botones de descargar y subir una nueva versión.

Por último, se simula una posible situación real creando una estructura de carpetas con contenido de diferente tipos para poder realizar pruebas de todas las acciones ejecutables de la aplicación y se refactoriza y completa la documentación del código.

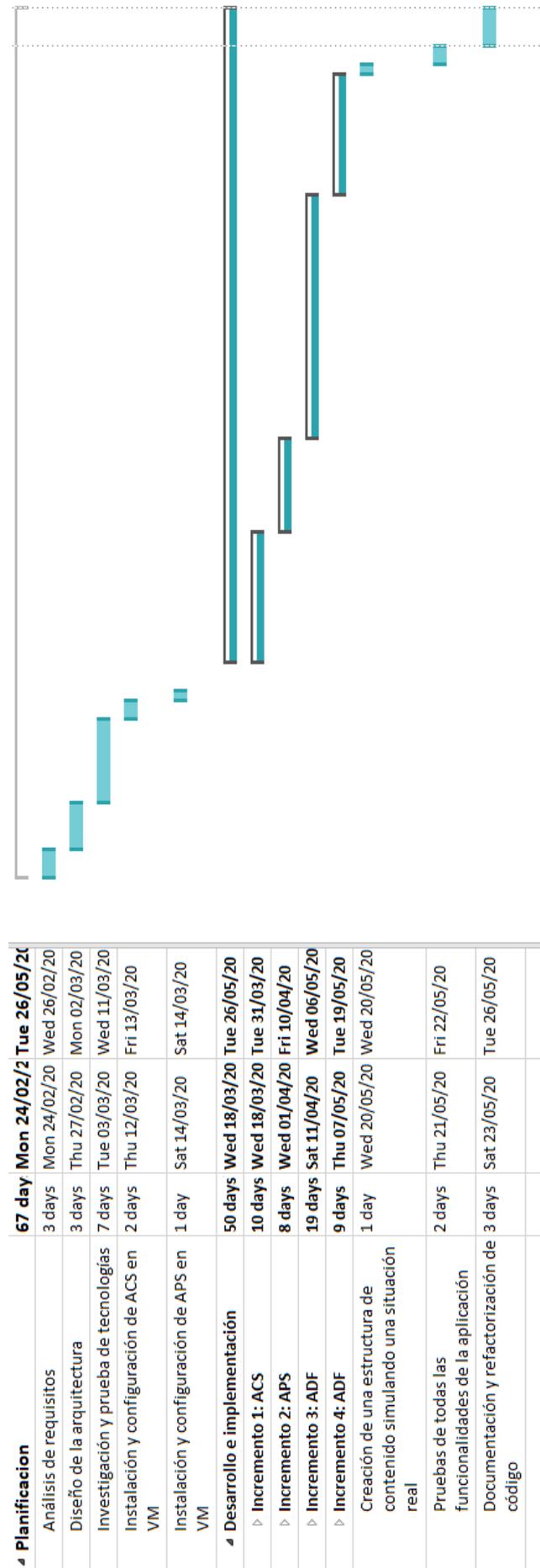


Figura 2: Diagrama de Gantt de la planificación del trabajo.

4 Presentación del problema y especificación de requisitos

4.1. Presentación del problema

El problema que se pretende resolver con este proyecto es salvar tiempo a cualquier persona que trabaje con documentos en idiomas diferentes al suyo, como los trabajadores de la consultoría informática Incentro mencionados en el caso de ejemplo presentado en el primer capítulo.

4.2. Análisis de requisitos

Los requisitos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación [23].

4.2.1. Actores del sistema

Los actores del sistema son los diferentes usuarios que interactuarán con la aplicación. A continuación se definen los tres tipos de usuario que utilizarán la aplicación:

Tabla 1: Actores del sistema

Actor	Descripción
Administrador	Tiene permiso para ejecutar todas las acciones posibles sobre el contenido de los sitios.
Colaborador	Tiene permiso para ejecutar todas las acciones posibles sobre el contenido de los sitios que sea de su propiedad. Puede editar el contenido creado por otros usuarios pero nunca eliminarlo.
Editor	Además de actuar como un colaborador, podrá realizar la tarea de revisión de la traducción.

4.2.2. Requisitos funcionales

Los requisitos funcionales (RF) describen las funciones que el software tiene que cumplir [24].

Tabla 2: Requisitos funcionales

ID	Descripción
RF01	Se mostrarán los sitios creados.
RF02	Se mostrará el contenido de los sitios.
RF03	Todos los usuarios podrán visualizar el contenido, tanto documentos como vídeos.
RF04	Todos los usuarios podrán traducir los archivos de tipo txt y docx.
RF05	Todos los usuarios podrán descargar el contenido de los sitios.
RF06	Todos los usuarios podrán copiar el contenido de los sitios a una localización diferente.

Continúa en la siguiente página.

Tabla 2 – Continuación de la página anterior.

ID	Descripción
RF07	Todos los usuarios podrán mover el contenido de los sitios a una localización diferente.
RF08	Todos los usuarios podrán eliminar el contenido de los sitios que haya sido creado o modificado por ellos.

4.2.3. Requisitos no funcionales

Los requisitos no funcionales (RNF) son los que limitan la solución. Se pueden clasificar de acuerdo a si son requerimientos de rendimiento, de mantenibilidad, de seguridad o de compatibilidad. [24]

Tabla 3: Requisitos no funcionales

ID	Descripción	Categoría	Importancia
RNF01	El acceso al sistema será a través de correo electrónico y contraseña.	Seguridad	Alta
RNF02	La interfaz del sistema tendrá menús fáciles de reconocer e intuitivos para el usuario.	Usabilidad	Media
RNF03	Los colores de los elementos de la aplicación deberán ser adecuados para una correcta visibilidad.	Usabilidad	Media
RNF04	El sistema debe mostrar mensajes de error que sean informativos y orientados al usuario final.	Usabilidad	Media
RNF05	El tiempo de aprendizaje sobre el sistema por parte del usuario deberá ser menor a 1 hora.	Usabilidad	Media
RNF06	Toda funcionalidad del sistema debe responder al usuario en menos de 5 segundos.	Eficiencia	Media

5 Especificación de casos de uso

En este capítulo se describe el funcionamiento general de la aplicación. Para esto se utilizan distintos diagramas de casos de uso que muestran las funciones y las diferentes interacciones de cada actor con el sistema. Además, se describen con más detalle los casos de uso más relevantes.

5.1. Diagramas de casos de uso

Seguidamente, se detallan los diferentes casos de uso del sistema. Estos están divididos según el tipo de usuario del sistema.

5.1.1. Diagrama para el rol Administrador

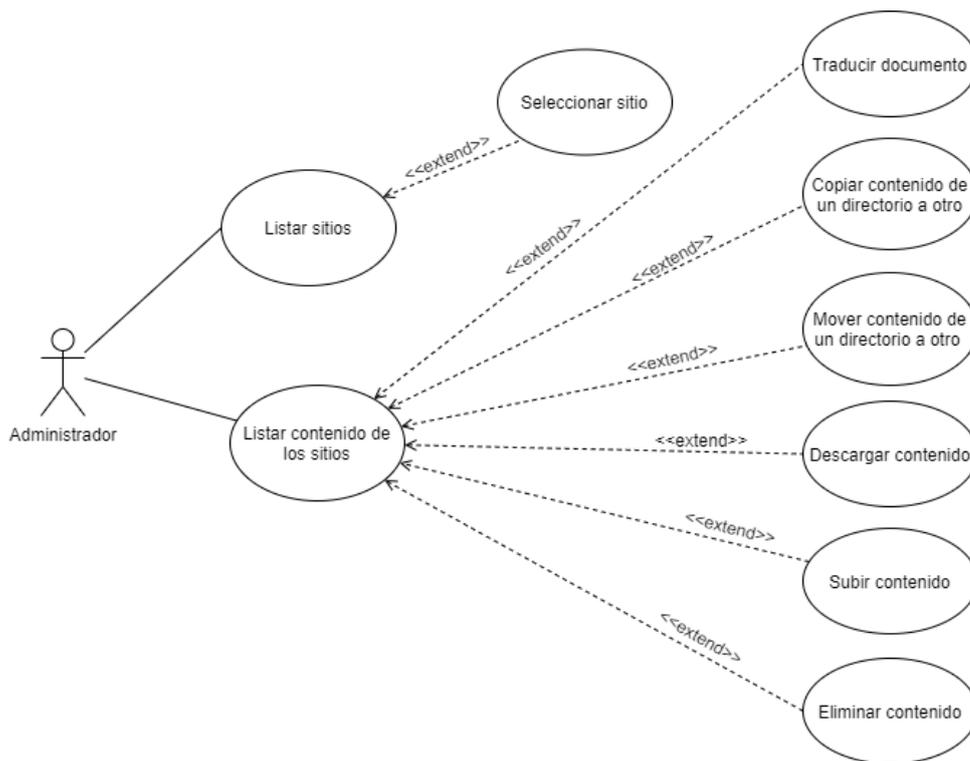


Figura 3: Diagrama de casos de uso para el rol Administrador.

5.1.2. Diagrama para el rol Colaborador

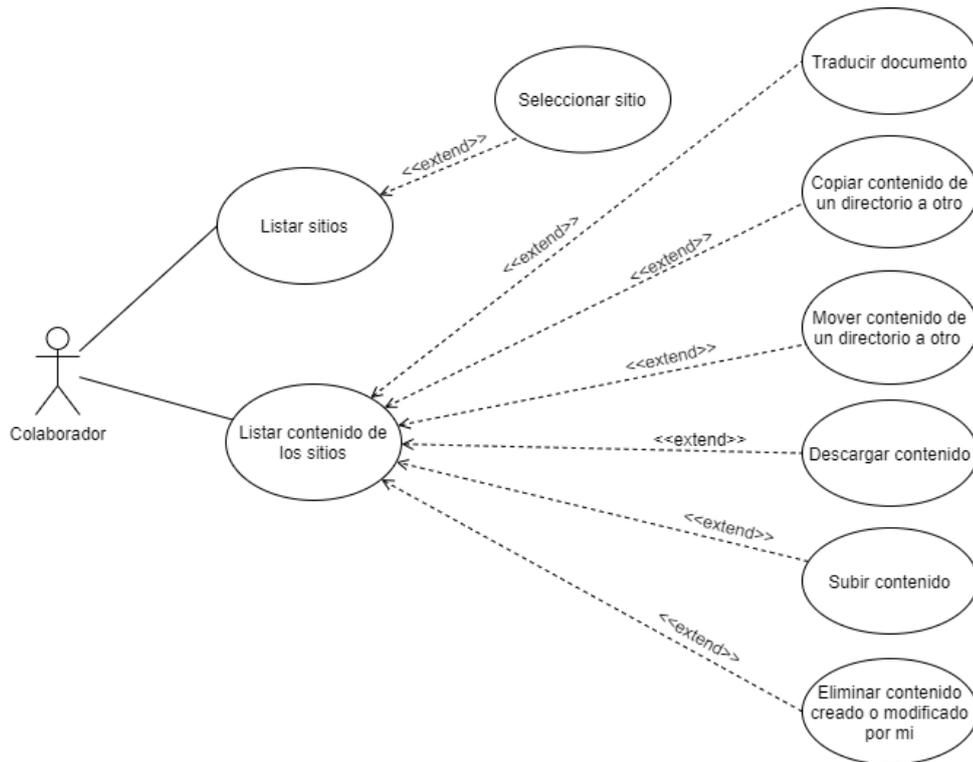


Figura 4: Diagrama de casos de uso para el rol Colaborador.

5.1.3. Diagrama para el rol Editor

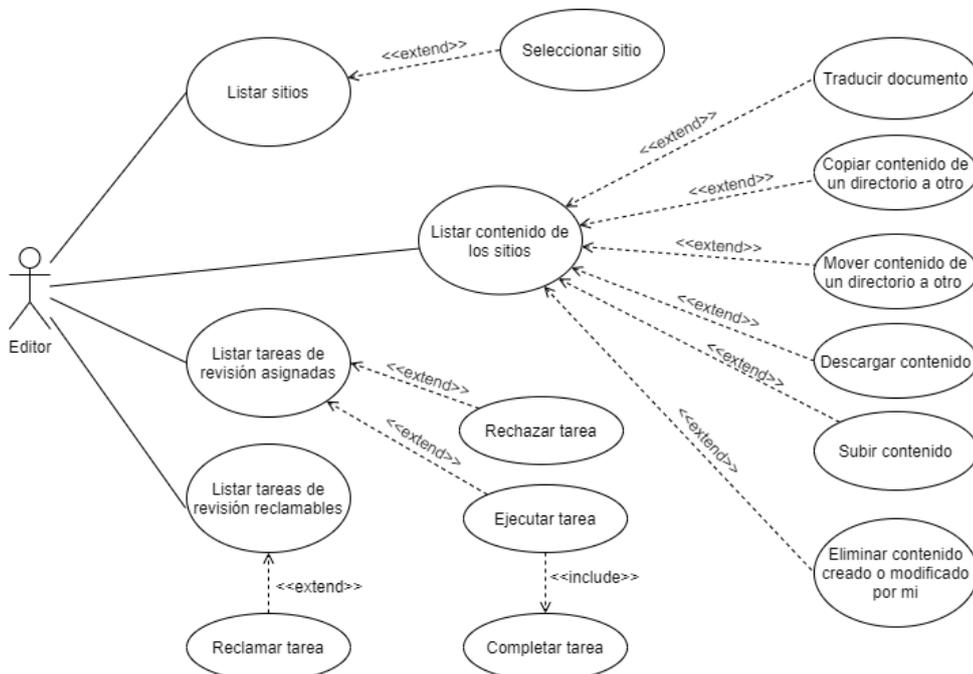


Figura 5: Diagrama de casos de uso para el rol Editor.

5.2. Plantillas de casos de uso

Para un mayor entendimiento de los casos de uso previamente definidos en los diagramas, a continuación se explica el comportamiento de los más relevantes mediante tablas.

Tabla 4: Plantilla Inicio de sesión.

ID	P01
Caso de uso	Inicio de sesión
Descripción	El usuario inicia sesión en el sistema
Actores	Todos
Precondiciones	El usuario tiene que tener una cuenta en ACS y en APS
Flujo principal	<ol style="list-style-type: none"> 1. El usuario introduce su correo electrónico y contraseña en los dos campos mostrados por el sistema. 2. El sistema carga la página con la lista de sitios.
Flujo alternativo	2.1 El sistema muestra un mensaje indicando un error en el inicio de sesión.

Tabla 5: Plantilla Listar sitios.

ID	P02
Caso de uso	Listar sitios
Descripción	El usuario visualiza la lista con los sitios que se han creado en ACS
Actores	Todos
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Contenido” del menú lateral. 2. El sistema muestra una lista con los sitios que están creados en ACS.
Flujo alternativo	2.1 El sistema muestra un mensaje indicando que no hay sitios creados.

Tabla 6: .Plantilla Listar contenido de los sitios

ID	P03
Caso de uso	Listar contenido de los sitios
Descripción	El usuario visualiza el contenido que existe en el sitio elegido
Actores	Todos
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona uno de los sitios mostrados en un desplegable. 2. El sistema muestra una lista con el contenido del sitio.
Flujo alternativo	2.1 El sistema muestra un mensaje indicando que la carpeta está vacía y que se arrastren ficheros para subirlos a la carpeta.

Tabla 7: Plantilla Listar tareas asignadas.

ID	P04
Caso de uso	Listar tareas de revisión asignadas
Descripción	El usuario visualiza la lista con las tareas de revisión de la traducción que tiene asignadas
Actores	Editor
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Tareas” del menú lateral. 2. El usuario selecciona la pestaña “Asignadas a ti” de la lista de tareas. 3. El sistema muestra una lista con tareas que el usuario tiene asignadas.
Flujo alternativo	3.1 El sistema muestra un mensaje indicando que no hay tareas.

Tabla 8: Plantilla Listar tareas reclamables.

ID	P05
Caso de uso	Listar tareas de revisión reclamables
Descripción	El usuario visualiza la lista con las tareas de revisión de la traducción que puede reclamar
Actores	Editor
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Tareas” del menú lateral. 2. El usuario selecciona la pestaña “Tareas reclamables” de la lista de tareas. 3. El sistema muestra una lista con tareas que el usuario puede reclamar.
Flujo alternativo	3.1 El sistema muestra un mensaje indicando que no hay tareas.

Tabla 9: Plantilla Traducir documento.

ID	P06
Caso de uso	Traducir documento
Descripción	El usuario selecciona la opción “Traducir” para traducir el documento al idioma que quiera
Actores	Todos
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona el botón “Traducir”. 2. El sistema muestra un diálogo para elegir el idioma al que traducir el documento. 3. El usuario elige uno de los idiomas mostrados. 4. El usuario clica en el botón “Empezar” para que se realice la traducción. 5. El sistema empieza el proceso de traducción.
Flujo alternativo	<ol style="list-style-type: none"> 2.1 El sistema muestra un mensaje indicando que el elemento que se ha seleccionado para traducir no tiene el formato correcto (txt o docx). 5.1 El sistema muestra un mensaje de error porque ese documento ya ha sido traducido al idioma seleccionado y no se puede crear un documento con el mismo nombre.

6 Creación de los entornos de trabajo

Antes de comenzar con la implementación del sistema, se han creado y configurado debidamente los servidores locales de desarrollo en los que trabajar. Esto consiste en dos máquinas virtuales, una con la instalación de ACS y la otra con APS, ambas con una distribución Ubuntu Server 18.04.2 de 64 bits.

- Servidor para ACS

En este servidor se utiliza la versión 5.2.6 de Alfresco Content Services. A través de la IP (en este caso es 192.168.1.155, pero puede variar al cambiar de conexión wifi) y usando el puerto 8080, se puede acceder a la interfaz de usuario por defecto (192.168.1.155:8080/share) y al servidor principal (192.168.1.155:8080/alfresco). La interfaz se utiliza en el proyecto para la creación y gestión de sitios, contenido y usuarios.

Para poder desarrollar extensiones para ACS, como se hace más adelante para la traducción, se utiliza un proyecto creado a partir del arquetipo de Maven alfresco-allinone-archetype. Este arquetipo permite tener un proyecto All-in-One en el que poder modificar tanto el módulo de Alfresco como el de Share. Para desplegar el módulo de Alfresco, se lleva el archivo tfg-acs-platform-jar-1.0-SNAPSHOT.jar al directorio /opt/alfresco-content-services/tomcat/webapps/alfresco/WEB-INF/lib y para el módulo del Share se lleva tfg-acs-share-jar-1.0-SNAPSHOT.jar a /opt/alfresco-content-services/tomcat/webapps/share/WEB-INF/lib.

- Servidor para APS

En este servidor se utiliza la versión 1.9.0.5 de Alfresco Process Services. Al igual que en ACS, a través de la IP (en este caso es 192.168.1.195 pero puede variar al cambiar de conexión wifi) y utilizando el puerto 8080, se puede acceder a la interfaz de usuario (192.168.1.195:8080/activiti-app) y al administrador de Alfresco Activiti (192.168.1.195:8080/activiti-admin). Desde la interfaz se crean los procesos y se interactúa con ellos, los usuarios pueden reclamar y completar las tareas de los procesos comenzados desde ahí.

- Creación de la aplicación de ADF y conexión con ACS y APS

Para poder configurar un desarrollo de ADF, es necesario tener node.js, Yeoman y un editor de código que en este caso es Visual Studio Code. Yeoman crea un proyecto e instala todas las dependencias requeridas por la aplicación. El proyecto se ejecuta en un terminal con el comando **npm start** y automáticamente se abre en el navegador la aplicación con la URL <http://localhost:4200>. Para que la aplicación pueda acceder al contenido e información de ACS y APS, es necesario indicarlo en el archivo `app.config.json` del proyecto de ADF. En este archivo se indican cuáles van a ser los proveedores y las direcciones de estos. Para decir que ambos van a ser los proveedores se le indica con `"providers":"ALL"`. Para las direcciones de cada uno se utiliza `"ecmHost":"http://192.168.1.155:8080"` para ACS y `"bpmHost":"http://192.168.1.195:8080"` para APS.

7 Diseño e implementación

En este apartado se explica el diseño utilizado en diferentes niveles y la implementación de cada una de las capas que componen la arquitectura del sistema.

7.1. Diseño arquitectónico

El diseño arquitectónico representa la estructura de los datos y de los componentes del programa que se requieren para construir un sistema. Considera el estilo de arquitectura que adoptará el sistema, la estructura y las propiedades de los componentes que lo constituyen y las interrelaciones que ocurren entre sus componentes arquitectónicos [25].

La arquitectura elegida para este sistema ha sido la arquitectura en tres capas. Este patrón consiste en dividir las funcionalidades del sistema en diferentes partes. Las tres capas son: presentación, negocio y datos. Esta separación permite realizar cambios en las capas sin afectar completamente al resto del sistema.

La capa de presentación corresponde con la interfaz desarrollada en ADF. La capa de negocio está compuesta por el desarrollo realizado en ACS y el flujo de trabajo realizado en APS. Las bases de datos de ACS y APS conforman la capa de datos.

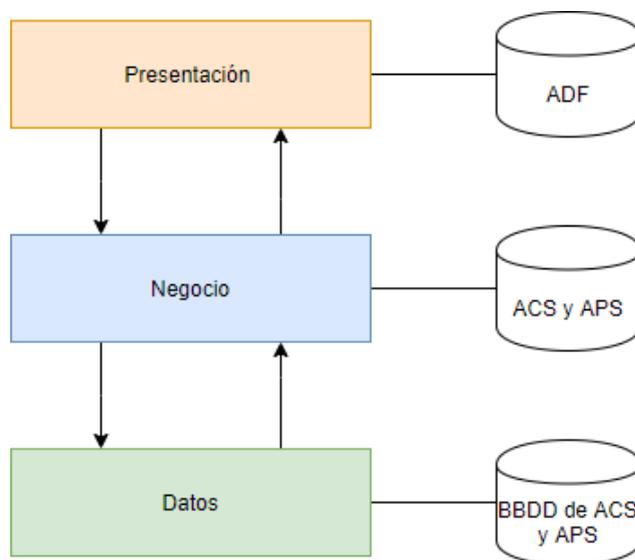


Figura 6: Arquitectura del sistema.

7.2. Capa de presentación

Esta es la capa en la que se lleva a cabo la mayor parte de la interacción entre el usuario y el sistema. Proporciona una interfaz gráfica al usuario en la que poder realizar todas las acciones descritas en los casos de uso. Aquí se presentan las diferentes vistas que componen la aplicación; estas se dividen en tres: vista de la página de inicio de sesión, vista de la página de contenido y vista de la página de tareas.

7.2.1. Vista de la página de inicio de sesión

El inicio de sesión en la aplicación se realiza autenticando al usuario en ACS y en APS con las mismas credenciales, esto se explicará más en el apartado de la capa de negocio.

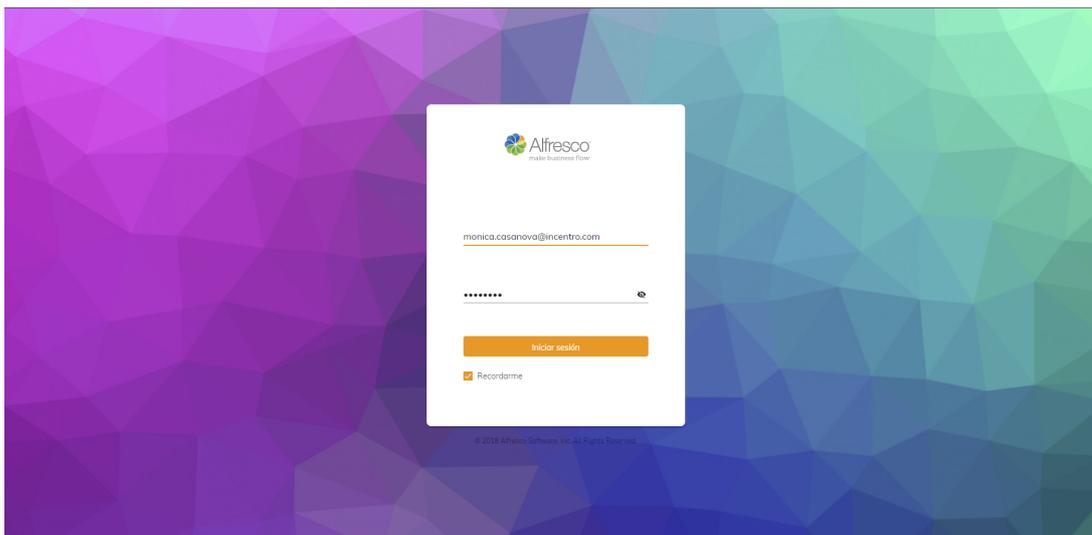


Figura 7: Página de inicio sesión con los datos del usuario rellenos.

7.2.2. Vista de la página de contenido

- Vista de los sitios con el menú lateral desplegado

Una vez iniciada la sesión, la primera página con la que se encuentra el usuario es la que contiene la lista de los sitios que han sido creados en ACS. Los sitios [26] son unidades organizativas en las que se pueden incluir directorios y archivos. La persona que crea el sitio es el administrador del mismo, con opción a que sea reemplazado. Estos pueden ser públicos, moderados o privados. Un sitio público es accesible para todos los usuarios pero solo los que sean miembros pueden trabajar con el sitio; cualquier usuario puede hacerse miembro. Un sitio moderado es accesible por todos los usuarios pero solo sus miembros pueden trabajar con los contenidos; los usuarios deben solicitar su incorporación al sitio. Un sitio privado es accesible únicamente por sus miembros y solo el administrador puede hacer miembros a los usuarios.

Esta página contiene un desplegable para elegir el sitio al que se desea acceder y ver directamente su contenido. También es posible el desplazamiento al contenido del sitio haciendo clic en la carpeta del sitio y ahí entrar en el directorio “documentLibrary”. Este directorio existe en todos los sitios y es donde se almacena todo el contenido.

- Vista del contenido de un sitio y las acciones posibles

El contenido de los sitios se divide en dos grandes tipos: los directorios y los archivos. Según sean de un tipo o de otro se pueden realizar diferentes acciones. Estas acciones se muestran en un

menú desplegable en cada elemento. Los directorios se pueden copiar y mover a otros directorios; también es posible descargar el directorio completo comprimido en un archivo ZIP y eliminarlo. Sobre los archivos se pueden realizar las mismas acciones, añadiendo la de traducir y visualizar. Para visualizar el contenido, ya sea un vídeo o un documento, simplemente hay que hacer doble clic sobre el archivo. Para la acción “Subir contenido” definida en los casos de uso no existe un botón sino que se realiza arrastrando el directorio o archivo deseado a donde se quiera subir. Solo se pueden subir directorios que no estén vacíos.

En la siguiente figura también se muestra el menú lateral, que por defecto está plegado en todas las vistas, y con el que se puede navegar entre la vista de contenido y la de tareas además de cerrar sesión en la aplicación

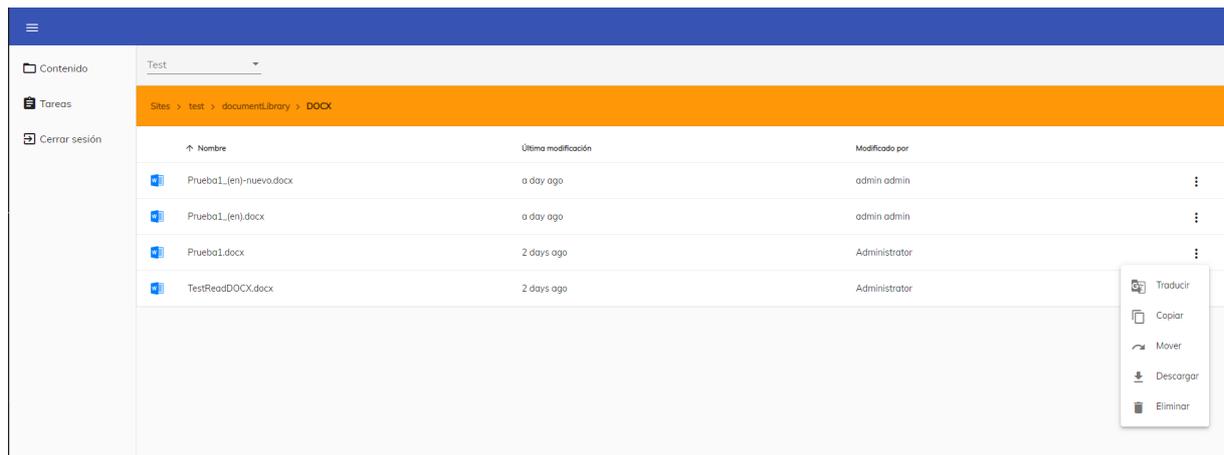


Figura 8: Acciones sobre los documentos.

La acción de traducir solo es posible para los documentos de extensión txt o docx, como ya se ha expuesto anteriormente. Esta acción se muestra en todo tipo de contenido pero a la hora de intentar ejecutarla, si el contenido no tiene las extensiones mencionadas, se muestra un mensaje indicando que no es posible traducir ese contenido. Si se ejecuta esta acción en uno de los documentos posibles, se muestra el siguiente diálogo que permite elegir el idioma al que se desea traducir y comenzar el proceso definido en APS.

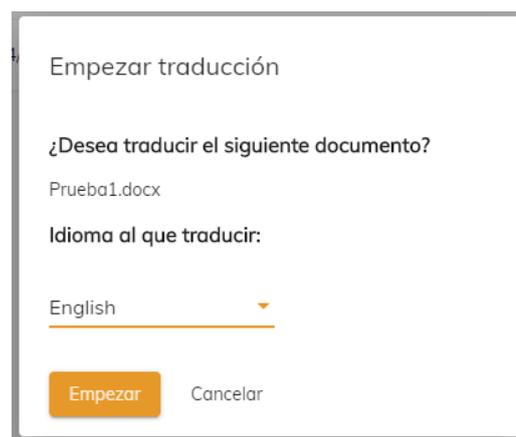


Figura 9: Menú para traducir con el idioma seleccionado.

7.2.3. Vista de la página de tareas

- Vista de la lista de tareas

En esta vista aparecen las tareas para revisar la traducción. En el momento en el que se realiza la traducción de un documento, una tarea para su revisión se crea en la pestaña “Tareas reclamables” para los usuarios editores. El editor tiene que reclamar la tarea para poder ejecutarla. Una vez reclamada, la tarea pasa a la pestaña “Asignadas a ti” y ahí se puede ejecutar y rechazar. En la tarea se muestra el nombre del documento original y el nombre del documento ya traducido, el cual es el mismo que el original con la abreviatura del idioma al que se ha traducido entre paréntesis. Para realizar la tarea se siguen dos acciones: primero, descargar el documento traducido y segundo, subir la nueva versión con las correcciones del editor; esta acción sustituye automáticamente el documento que se había traducido.

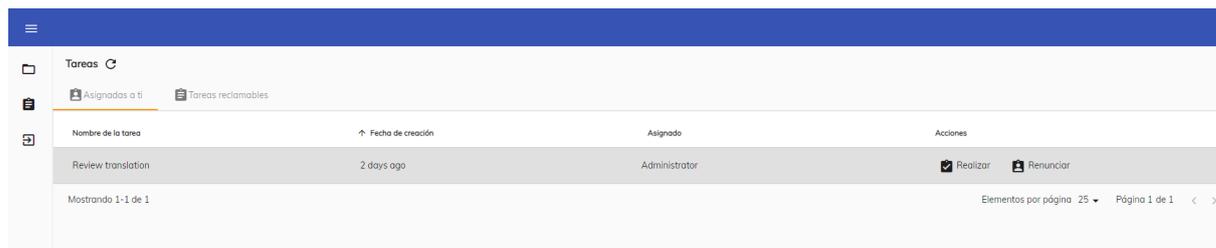


Figura 10: Lista de las tareas asignadas al usuario.

7.3. Capa de negocio

En esta capa se encuentra toda la funcionalidad de la aplicación. Responde a las solicitudes de la capa de presentación y obtiene los datos que necesita para procesar dichas solicitudes de la capa de datos. Como ya se ha indicado previamente, esta capa está compuesta por dos tecnologías: ACS y APS. Cada una de las vistas expuestas en el anterior capítulo interactúa con una tecnología o en algún caso con las dos. El inicio de sesión interactúa con ambas, la página de contenido se relaciona solo con ACS, a excepción de la acción “Traducir” que también interactúa con APS, y la página de tareas se comunica únicamente con APS.

- Inicio de sesión

Como ya se ha mencionado, en el inicio de sesión es necesario que se acredite al usuario tanto en ACS como en APS para poder utilizar las funcionalidades de ambos servicios una vez se está dentro de la aplicación. Los usuarios de ACS tienen nombre de usuario y correo electrónico pero para iniciar sesión se utiliza el nombre de usuario. Por otro lado, los usuarios en APS solo constan de correo electrónico, así que para que el inicio de sesión en la aplicación funcione correctamente contra ambas tecnologías es necesario que el nombre de usuario en ACS coincida con el correo electrónico del usuario registrado en APS. El componente *Login component* [27] acredita al usuario en ACS y APS por separado o en las dos a la vez. Para que esta acreditación sea posible, es necesario habilitar el CORS (*Cross-origin resource sharing*) tanto en ACS como en APS. CORS [28] es un mecanismo que utiliza cabeceras HTTP para decirle al navegador que dé acceso a una aplicación que se está ejecutando en un origen a los recursos de un origen diferente. Para habilitar este mecanismo y permitir que la aplicación web pueda acceder a los recursos de ACS hay que modificar en el `web.xml` en la máquina virtual de ACS el parámetro `cors.allowOrigin` para que permita el acceso al origen de la aplicación web (`localhost:4200`). En el caso de APS, el archivo que hay que modificar es el `activiti-app.properties` y la propiedad en la que se indican los orígenes permitidos es `cors.allowed.origins`.

- Página de contenido

La página de contenido está constituida por cinco componentes de ADF que acceden a la información de ACS. El primero es *Document List* [29], que obtiene todo el contenido del directorio que se le indique. El segundo, *Sites Dropdown* [30], muestra un desplegable con los sitios del usuario. El siguiente es *Content Action* [31], que permite realizar algunas de las acciones que ofrece ACS sobre los archivos y los directorios como descargar, copiar, mover y borrar o crear acciones personalizadas como en el caso de la acción “Traducir”. Después, se utiliza *Breadcrumb* [32] para indicar la posición actual en la jerarquía de carpetas. Y por último, el componente utilizado para la visualización del contenido es *Viewer* [33]; muestra el contenido a partir de su id y también permite personalizar una barra de herramientas para realizar acciones sobre ese contenido.

- Página de tareas

Esta página se compone de los componentes *Task List* [34] y *Task Details* [35]. El *Task List* renderiza una lista con las tareas asignadas y candidatas a ser reclamadas de los procesos que están empezados en APS. El *Task Details* se encarga de mostrar el formulario que se ha creado en la tarea de revisión de la traducción del proceso.

Aunque los componentes de ADF mencionados están implementados en la capa de presentación, estos acceden a las funciones de ACS y APS; por eso se habla de ellos en la capa de negocio.

7.3.1. Traducción de documentos

La traducción de los documentos se realiza en ACS. Se ha creado un web script que, dependiendo del tipo de documento, llamará a la clase `TranslateTxt.java` o a `TranslateDocx.java`. Estas clases son las encargadas de procesar el documento para que la API del traductor Google pueda traducirlo.

Un web script es una forma de implementar una API RESTful y también podría denominarse servicio web [36]. El cliente utiliza la API para comunicarse con el servicio web. Que la API sea RESTful significa que cumple con la arquitectura software REST (*representational state transfer*); esta arquitectura está conformada por una serie de principios:

- Utiliza un protocolo cliente-servidor sin estado. Esto significa que cada solicitud debe contener toda la información necesaria para realizar la petición ya que el servidor no tiene que memorizar el estado del cliente.
- El acceso a los recursos del servidor se realiza a través del URI (*uniform resource identifier*).
- Las operaciones disponibles son las mismas que los métodos de HTTP: GET, PUT, POST, DELETE son los más utilizados.
- Tiene una interfaz uniforme gracias a que se realizan acciones concretas sobre los recursos identificados con URI.
- Usa hipermedios en la respuesta para enlazar los recursos con otros recursos que puedan estar relacionados.
- El cliente se puede conectar al servidor a través de la interfaz o de un intermediario.

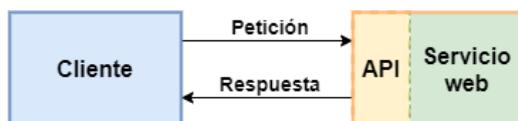


Figura 11: Interacción entre cliente y servicio web a través de una API.

El web script está formado por tres archivos: un descriptor, un controlador y una plantilla. Los tres archivos deben contener en su nombre el método HTTP que se utiliza. En este caso se realiza una petición POST porque se van a enviar datos al servidor para crear un nuevo recurso que será el documento traducido. En la siguiente figura se muestran los archivos que conforman el web script.

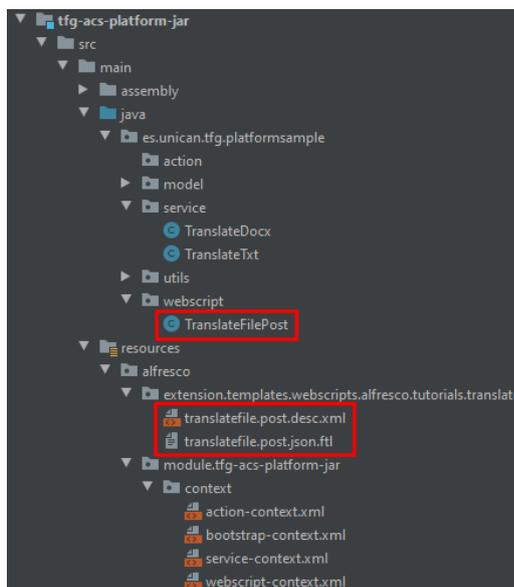


Figura 12: Archivos para el web script de traducción en el proyecto de ACS.

El descriptor (`translatefile.post.desc.xml`) define la URL que hay que utilizar para llamar al web script y el formato de respuesta que se espera al llamar a este web script. La URL que se define no es la de la llamada completa, es solo la parte que se añade a la URL que envía la petición al controlador (`http://{host}:{port}/alfresco/s`). Para identificar el web script de manera única se utiliza el nombre del descriptor, que sigue el siguiente convenio: `<web script id>.<método HTTP>.desc.xml`. En este caso “translatefile” será el identificador único del web script aunque el método HTTP también contribuye a su identificación.

```
<webscript>
  <description>Translate file webscript.</description>
  <url>/custom/translate-file</url>
  <authentication>user</authentication>
  <format default="json"/>
</webscript>
```

Figura 13: Descriptor del web script.

El controlador (`TranslateFilePost.java`) es el que se encarga de realizar la lógica de negocio. En este caso recibe el nombre del documento (con extensión incluida) que se quiere traducir, el UUID (*Universal Unique Identifier*) del directorio en el que está almacenado y el código ISO 639-1 del idioma al que se quiere traducir. En ACS todo el contenido es representado por nodos a los cuales se accede a través de una referencia (`NodeRef`). Esta referencia está formada por el protocolo de almacenamiento de contenido, el nombre del almacenamiento de contenido y el UUID del contenido como por ejemplo: `workspace://SpacesStore/ccb906ba-a768-4ccb-8b26-515119e1efdc`. El controlador completa el `NodeRef` del directorio en el que se encuentra el documento con el UUID que recibe y después utiliza el servicio `NodeService` [37], que permite gestionar los nodos, para obtener la referencia del documento en dicho directorio.

La plantilla (`translatefile.post.json.ftl`) recoge la respuesta que devuelve el web script. El nombre de este archivo, además de contener el método HTTP utilizado, tiene el formato de la respuesta que en este caso es un JSON. La respuesta de este web script son el nombre del documento que se quiere traducir y el nombre del documento una vez traducido.

Además de crear estos tres archivos, es necesario asociar el controlador con el web script definiendo un *Spring bean* en el archivo `webscript-context.xml`. El identificador del *bean* tiene que atenerse al formato `webscript.<package id>.<web script id>.<método HTTP>`.

```
<bean id="webscript.alfresco.tutorials.translate.translatefile.post"
      class="es.unican.tfg.platformsample.webscript.TranslateFilePost"
      parent="webscript">
</bean>
```

Figura 14: *Spring bean* que asocia el controlador con el web script

Como ya se ha mencionado anteriormente, para poder traducir los documentos, primero es necesario procesarlos. En el caso de los documentos txt, se utiliza el servicio *ContentService* [38] de ACS que permite acceder y manipular el contenido existente. Con este servicio se puede obtener el contenido de este documento como un *String* y enviárselo a la API del traductor de Google. Para poder utilizar esta API, primero se debe crear un proyecto en la *Google Cloud Console* y luego habilitar la API de Cloud Translation. Seguidamente, hay que crear una cuenta de servicio con la que se obtiene autorización para poder utilizar la API. Esta es un archivo JSON que contiene una clave privada. Dicha clave es referenciada desde la variable de entorno `GOOGLE_APPLICATION_CREDENTIALS` en la máquina virtual de ACS. Una vez hecho esto, se añaden al `pom.xml` las dependencias necesarias. En la siguiente figura se muestra cómo funciona la traducción del texto leído de un documento txt. En la petición se establece el tipo de contenido que va a traducir con `setMimeType("text/plain")`, el idioma al que se va a traducir con `setTargetLanguageCode(targetLanguage)` y el contenido a traducir con `addContents(textRead)`.

```
String projectId = "dynamic-nomad-272918";
try (TranslationServiceClient client = TranslationServiceClient.create()) {
    LocationName parent = LocationName.of(projectId, "global");
    TranslateTextRequest request =
        TranslateTextRequest.newBuilder()
            .setParent(parent.toString())
            .setMimeType("text/plain")
            .setTargetLanguageCode(targetLanguage)
            .addContents(textRead)
            .build();
    TranslateTextResponse response = client.translateText(request);
    for (Translation translation : response.getTranslationsList()) {
        translatedText += translation.getTranslatedText();
    }
}
```

Figura 15: Código de traducción de texto con la API de Google

Una vez realizada la traducción, se utiliza el servicio *FileFolderService* [39] para crear un nuevo nodo en el directorio con el nombre del archivo original y el código del idioma al que se ha traducido entre paréntesis (por ejemplo `Prueba_(en).txt`) y se escribe el texto traducido con este mismo servicio.

Por otro lado, para la traducción de documentos docx, en vez de extraer el texto directamente como en el otro caso, el documento se convierte de docx a XHTML. Esto permite que no se pierdan formatos como el color o tamaño de letra que tiene el documento original. Para hacer esta conversión se utiliza el convertidor de la librería Apache POI. Una vez realizada la conversión, se utiliza el mismo código de la Figura 15 cambiando el tipo a HTML con `setMimeType("text/html")`. El contenido

que recibe la API de Google es el documento XHTML en formato *String*. Por último, se convierte de vuelta el XHTML traducido a docx con la librería Docx4J, ya que Apache POI solo ofrece la opción de convertir un docx a XHTML y no al revés. Se han utilizado ambas librerías en vez de solo Docx4J porque con esta última en la conversión de docx a XHTML se perdía el tamaño de letra y el alineamiento del texto.

7.3.2. Proceso

El proceso para la traducción y revisión está conformado por cuatro elementos que se muestran en la siguiente figura:



Figura 16: Flujo de trabajo del proceso de traducción.

- Evento inicial

El evento inicial contiene un formulario en el que se recoge el nombre del documento que se desea traducir, el UUID del directorio en el que se encuentra el documento y el código del idioma al que se va a traducir. Estos son los datos que le llegan al web script para llevar a cabo la traducción. Este formulario se rellena desde el código de la acción “Traducir” de la aplicación web.

Figura 17: Formulario para comenzar el proceso.

- Translate file

Esta tarea es una tarea de llamada REST, se utiliza para comunicarse con el *endpoint* de la API que es la URL que se utiliza para realizar la petición. En esta tarea se configuran tres propiedades. La primera es definir el *endpoint* al que llamar, en este caso una llamada POST a la URL `http://192.168.1.155:8080/alfresco/s/custom/translate-file`, siendo 192.168.1.155 la IP del *host* que en este caso es la máquina virtual de ACS. Después, se rellena el cuerpo de la petición con los valores inicialmente completados en el formulario. Y por último, se mapea la respuesta con formato JSON del *endpoint* en variables del proceso, que en este caso son el nombre original del documento y el nombre del documento traducido.

- Tarea de usuario Review translation

La tarea de revisión es una tarea de usuario que se le asigna al grupo “Editores”. Este grupo se crea desde la gestión de usuarios que proporciona APS y se añaden ahí los usuarios que vayan a tener el rol de editor. Cuando un proceso llega a esta tarea, todos los usuarios pertenecientes a dicho grupo reciben un correo electrónico informándoles de que tienen una tarea de revisión que pueden reclamar. Para permitir que APS envíe correos hay que configurar el servidor de correo electrónico en `activiti-app.properties`.

```
email.enabled=true
email.host=smtplib.gmail.com
email.port=587
email.tls=true
email.useCredentials=true
email.username=monica.casanova@incentro.com
email.password=CONTRASEÑA
# The base url that will be used to create urls in emails.
email.base.url=http://192.168.1.195:8080
email.from.default=no-reply@activiti.alfresco.com
email.from.default.name=Activiti
email.feedback.default=aps@alfresco.com
```

Figura 18: Configuración del servidor de correo electrónico

La contraseña que se utiliza se tiene que generar a través de <https://myaccount.google.com/apppasswords> para evitar que haya problemas con la autenticación en dos pasos de Google. Una vez configurado, en la tarea de revisión se activa el envío de correos electrónicos y se indica que se utilice la plantilla que existe por defecto. Esta plantilla indica que se ha creado una tarea que el usuario puede reclamar.

El formulario asociado a la tarea muestra los nombres de los documentos recogidos en la respuesta de la API en la anterior tarea. Además, tiene dos campos personalizados los cuales son botones para descargar el documento traducido y subir una nueva versión de ese documento; estos dos botones solo se pueden ver y utilizar desde la aplicación ya que son creados como widgets del formulario en la implementación de la página de tareas. Para crear campos personalizados en formularios se utiliza una plantilla de formulario personalizada [40] que se basa en la plantilla por defecto y se pueden tanto eliminar campos que ya existen, reordenar, modificar o crear nuevos campos.

- Evento final

Este evento se encarga de finalizar la ruta actual de ejecución; como en este proceso solo hay un camino posible, es el final del proceso completo.

7.4. Capa de datos

La base de datos que utiliza ACS es postgresql; en ella se almacenan los metadatos del contenido que se crea, por ejemplo la fecha de creación o el tamaño. La gran mayoría de elementos que hay en ACS son nodos, y algunos nodos están formados exclusivamente por metadatos como por ejemplo los usuarios. El contenido de los documentos en sí se guarda en el sistema de ficheros. La base de datos que utiliza APS es h2; en ella se almacenan los usuarios y grupos creados, el proceso de traducción, las tareas del proceso y las variables que se utilizan en el proceso.

8 Pruebas y resultados

8.1. Pruebas unitarias

Las pruebas unitarias [24] verifican el funcionamiento de módulos de software de manera aislada como puede ser una clase o un método. Estas pruebas se han realizado en cada incremento a medida que se avanzaba dentro de este.

- Pruebas para el web script.

El web script creado se desarrolló en diferentes fases y se probaron cada una de estas a través de la herramienta Postman y comprobando al mismo tiempo los *logs* de ACS a través del fichero *catalina.out*. Lo primero de todo había que comprobar era que el web script fuese accesible a través de la URL definida y que recogía el cuerpo de la petición correctamente.



Figura 19: URL y cuerpo en formato JSON de la petición en Postman.

Una vez probada la llamada, se utilizó la respuesta del web script para ir probando que los diferentes métodos para procesar y traducir los documentos devolvían resultados correctos. Estos son la lectura del contenido del documento txt, la traducción del texto, la conversión de docx a XHTML y su traducción.

- Pruebas para el proceso.

En el proceso había que probar que las tareas funcionasen correctamente y que el flujo se completase con éxito. Lo primero, que la tarea de llamada REST realizaba sin error la llamada, para esto se utilizó la opción “Prueba” que ofrece la propia tarea que te permite introducir los campos requeridos a mano y te muestra la respuesta en caso de éxito o el error. Después había que comprobar que el formulario de la tarea de revisión muestra los valores que se recogen en la respuesta de la anterior tarea. Para esto se ejecuta el proceso desde el *activiti-app* de APS y se comprueba que el proceso se puede completar exitosamente.

8.2. Pruebas de integración

Un aspecto importante en el desarrollo del sistema es la integración de las funcionalidades implementadas en cada uno de los incrementos. Al finalizar cada incremento, a excepción del primero, se ha comprobado que los nuevos avances implementados funcionasen con lo desarrollado en el anterior incremento de la siguiente forma:

- Integración de los incrementos 1 y 2.

En esta integración se tenía que comprobar que la API desarrollada en ACS y el proceso de APS funcionaban conjuntamente. Esta prueba también se ha realizado en el apartado de pruebas unitarias ya que es imposible que el proceso funcione correctamente si falla la tarea de llamada REST.

- Integración de los incrementos 2 y 3.

Esta prueba se centraba en comprobar que al ejecutar la acción “Traducir” de un documento en la aplicación, la primera parte del proceso (es decir, el evento inicial y la tarea de llamada REST “Translate file”) se ejecutaban correctamente. Para ello había que confirmar que los parámetros que se le enviaban al formulario del evento inicial eran correctos y que se mostraba el documento traducido en la lista de contenido.

- Integración de los incrementos 2 y 4.

En esta integración se ha verificado que la tarea de revisión de la traducción del proceso aparecía correctamente en la lista de tareas de la aplicación y que al completarla el proceso terminaba. Esto se ha comprobado a través del `activiti-admin` de APS que permite controlar qué punto del proceso se está ejecutando y si se ha terminado.



Figura 20: Ejecución de una instancia del proceso que está en la tarea Review translation.

- Integración de los incrementos 3 y 4.

Esta integración fue sencilla porque no existen dependencias entre estos incrementos. Así que simplemente se comprobó que el cambio de la página de contenido a la de tareas funcionaba.

8.3. Pruebas de sistema

8.3.1. Pruebas de rendimiento

Para las pruebas de rendimiento se buscó cumplir con el requisito RNF06. Se comprobó que el tiempo definido no se sobrepasaba en las acciones implementadas utilizando el temporizador que crea la función `console.time()` en TypeScript. La funcionalidad que mayor tiempo de carga tiene es la de visualizar contenido. Esta funcionalidad carga el contenido al instante excepto en el caso de que sea nuevo. Por ejemplo, al visualizar un documento recién traducido y que no se ha visualizado previamente, alfresco realiza una conversión del documento a pdf para poder visualizarlo y por eso el documento tarda en cargar bastante.

8.3.2. Pruebas de usabilidad

Estas pruebas se corresponden con los requisitos RNF02, RNF03, RNF04 y RNF05 en los que se recogen las características para facilitar el aprendizaje y la interacción del usuario con el sistema. Para probar que dichos requisitos se cumplían, se entregó la aplicación a compañeros de la empresa. De estas pruebas se concluyeron dos mejoras que ayudarían al usuario. La primera, que la interacción con la lista de tareas mostrase más información sobre la tarea de revisión, como por ejemplo el nombre del documento que se ha traducido. La segunda, que la subida de directorios y archivos se pueda realizar también a través de un botón, ya que el mensaje de subir los archivos arrastrándolos solo se muestra en directorios vacíos.

8.4. Pruebas de aceptación

Las pruebas de aceptación se han realizado en cada incremento descrito en la planificación y al final del desarrollo. El co-director del proyecto se aseguró que los requisitos definidos en cada incremento se cumplían, verificó que los objetivos definidos al principio se completaron y realizó las pruebas finales sobre el sistema comprobando que todas las funciones de la aplicación se ejecutaban correctamente.

9 Conclusiones y trabajos futuros

En este último capítulo se valoran tanto los objetivos inicialmente establecidos y en qué medida se han cumplido como los fallos o carencias encontradas (que dan paso a las posibles extensiones o mejoras para un futuro).

Este proyecto tenía como objetivo principal el desarrollo de una aplicación web que ofreciese la posibilidad de traducir documentos y posteriormente revisar dicha traducción. Dicho objetivo se cumplió y además el resto de objetivos y todos los requisitos establecidos que añadían diferentes funciones a la aplicación y hacían de esta una aplicación más atractiva.

La mayor dificultad con la que me he encontrado en la realización del proyecto ha sido el procesamiento de los documentos para la traducción, siendo este la lectura del contenido y la escritura del nuevo documento. La dificultad del procesamiento de documentos docx fue mayor que en el caso de los documentos txt debido a que el texto que contiene los primeros tiene formato, como por ejemplo distintos tamaños de letra.

La traducción de documentos completos no es algo nuevo. Google Drive y Microsoft Word ya cuentan con herramientas para traducir sus documentos. Lo que aporta esta aplicación es que recoge características de ambas tecnologías. Por una parte, Microsoft Word permite traducir los documentos de extensión docx y Google Drive permite la traducción de sus documentos docs. Por otro lado, Google Drive ofrece un servicio de almacenamiento en la nube. Así se ha combinado la posibilidad de almacenar los archivos en la nube y traducir los documentos de extensión docx y txt. Además, la opción de traducir documentos txt, sin cambiar la extensión del documento, no está entre lo que ofrecen Microsoft Word y Google Drive.

Personalmente, el proyecto me ha permitido poner a prueba muchos de los conocimientos adquiridos durante la carrera y durante mi estancia en la empresa, además de adquirir nuevos conocimientos y enfrentarme al reto de realizar un desarrollo software desde cero pasando por todas sus fases.

Aunque los objetivos establecidos se hayan cumplido satisfactoriamente, se puede seguir mejorando el sistema, tanto para introducir nuevos elementos como para mejorar los que ya están implementados.

Una de las posibles mejoras consiste en sustituir la interfaz que proporciona ACS (Share) con la aplicación de ADF y así poder realizar toda o la mayor parte de la gestión desde una sola aplicación. Para conseguir esto sería necesario añadir más elementos y funciones: poder crear y gestionar los sitios, crear contenido y realizar más acciones sobre este como editar sus propiedad (cambiar nombre) y crear y gestionar los usuarios. La función de creación de usuarios tendría que encargarse de crear el usuario tanto en ACS como en APS y así se evitaría tener que crear los usuarios dos veces. Además, se añadiría una barra para buscar contenido.

Otras funciones que añadir, que se plantearon pero no se realizaron por falta de tiempo, son una barra de búsqueda para poder encontrar los archivos deseados más rápidamente y una opción para cambiar el idioma en el que se muestra la aplicación. Esta última opción está implementada en parte, ya que todos los textos (excepto la lista de idiomas a los que traducir que solo están en inglés) que se muestran están definidos en español e inglés en el proyecto de ADF y se puede cambiar de un idioma a otro modificando la siguiente línea de código `this.translationService.use('es');`.

Por otro lado, una gran mejora para la aplicación sería poder editar los documentos en la propia tarea de revisión de manera online y así evitar que el usuario tenga que descargarse el documento y utilizar otro programa en su ordenador para editar el documento. Esta mejora se planteó al principio del proyecto como algo adicional en caso de que hubiese tiempo y posibilidad de realizarlo.

En cuanto al proceso, este siempre puede ampliarse para introducir nuevas tareas que tengan que realizar los usuarios o modificar los formularios ya existentes. Por ejemplo, sería posible añadir más tareas de revisión y diferentes grupos de editores para que el documento lo revisen varias personas y la calidad de la traducción mejore.

Por último, una última mejora que se intentó realizar pero no se consiguió fue que en la traducción de un documento docx no se perdiesen ningún estilo que tenía el documento original. Como ya se mencionó antes, el color y tamaño de letra se mantienen igual pero unos cambios que ocurren en la conversión es la modificación de la fuente de la letra y el desalineamiento de las imágenes.

10 Bibliografía

- [1] Alfresco Software Inc. Software de gestión de contenido empresarial: preguntas frecuentes. ¿qué es el software de gestión de contenido empresarial? <https://www.alfresco.com/es/ecm-software>. Accedido el 2020-5-18.
- [2] Alfresco Software Inc. Activiti. <https://www.activiti.org/>. Accedido el 2020-06-22.
- [3] Alfresco Software Inc. Alfresco Process Services (BPM). <https://www.alfresco.com/es/bpm-software>. Accedido el 2020-05-18.
- [4] Alfresco Software Inc. Alfresco Application Development Framework. <https://www.alfresco.com/es/ecm-software/application-development-framework>. Accedido el 2020-05-18.
- [5] Mark Massé. REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces. O'Reilly, 2011.
- [6] Angular Team at Google. Introduction to the Angular Docs. <https://angular.io/docs>. Accedido el 2020-05-19.
- [7] Google Cloud. Traduce texto (básico). <https://cloud.google.com/translate/docs/basic/translating-text>. Accedido el 2020-05-18.
- [8] Oracle. Chapter 1. First Steps. <https://www.virtualbox.org/manual/ch01.html>. Accedido el 2020-05-18.
- [9] MobaXterm. Presentation. https://mobaxterm.mobatek.net/documentation.html#1_1. Accedido el 2020-05-18.
- [10] Jet Brains. IntelliJ IDEA overview. <https://www.jetbrains.com/help/idea/discover-intellij-idea.html>. Accedido el 2020-05-18.
- [11] Microsoft. Getting Started. <https://code.visualstudio.com/docs>. Accedido el 2020-05-18.
- [12] Git. <https://git-scm.com>. Accedido el 2020-06-07.
- [13] Atlassian. Introducing Sourcetree for Windows – a free desktop client for Git. <https://www.atlassian.com/blog/archives/introducing-sourcetree-git-client-microsoft-windows>. Accedido el 2020-06-07.
- [14] Postman. What is Postman? <https://www.postman.com/>. Accedido el 2020-05-25.
- [15] OpenJS Foundation. Acerca de Node.js. <https://nodejs.org/es/about/>. Accedido el 2020-06-21.
- [16] Yeoman. What's Yeoman? <https://yeoman.io/>. Accedido el 2020-06-21.
- [17] The Apache Software Foundation. Welcome to Apache Maven. <https://maven.apache.org/>. Accedido el 2020-06-21.

- [18] Oracle. Conozca más sobre la tecnología Java. <https://www.java.com/es/about/>. Accedido el 2020-05-18.
- [19] W3schools. HTML introduction. https://www.w3schools.com/html/html_intro.asp. Accedido el 2020-05-18.
- [20] Martin Caceres. ¿qué es Typescript? <https://devcode.la/blog/que-es-typescript/>. Accedido el 2020-06-21.
- [21] Microsoft. Typescript. <https://www.typescriptlang.org/index.html>. Accedido el 2020-05-19.
- [22] W3schools. CSS tutorial. <https://www.w3schools.com/css/>. Accedido el 2020-05-19.
- [23] Ian Sommerville. Software Engineering. Pearson Educación, 9a. edition, 2011.
- [24] P. Bourque and R.E. Fairley, editors. Guide to the Software Engineering Body of Knowledge. IEEE Computer Society, version 3.0 edition, 2014.
- [25] Roger S. Pressman. Ingeniería del software. Un enfoque práctico. McGraw-Hill, 7 ed. edition, 2010.
- [26] Alfresco Software Inc. Sitios. <https://docs.alfresco.com/using-es/concepts/sites-intro.html>. Accedido el 2020-06-11.
- [27] Alfresco Software Inc. Login component. <https://www.alfresco.com/abn/adf/docs/core/components/login.component/>. Accedido el 2020-06-16.
- [28] Mozilla and individual contributors. Cross-Origin Resource Sharing (CORS). <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>. Accedido el 2020-06-16.
- [29] Alfresco Software Inc. Document List component. <https://www.alfresco.com/abn/adf/docs/content-services/components/document-list.component/>. Accedido el 2020-06-11.
- [30] Alfresco Software Inc. Sites Dropdown component. <https://www.alfresco.com/abn/adf/docs/content-services/components/sites-dropdown.component/>. Accedido el 2020-06-11.
- [31] Alfresco Software Inc. Content Action component. <https://www.alfresco.com/abn/adf/docs/content-services/components/content-action.component/>. Accedido el 2020-06-11.
- [32] Alfresco Software Inc. Breadcrumb component. <https://www.alfresco.com/abn/adf/docs/content-services/components/breadcrumb.component/>. Accedido el 2020-06-18.
- [33] Alfresco Software Inc. Viewer component. <https://www.alfresco.com/abn/adf/docs/core/components/viewer.component/>. Accedido el 2020-06-18.
- [34] Alfresco Software Inc. Task List component. <https://www.alfresco.com/abn/adf/docs/process-services/components/task-list.component/>. Accedido el 2020-06-11.
- [35] Alfresco Software Inc. Task Details component. <https://www.alfresco.com/abn/adf/docs/process-services/components/task-details.component/>. Accedido el 2020-06-11.
- [36] Alfresco Software Inc. Web scripts. <https://docs.alfresco.com/5.2/references/dev-extension-points-webscripts.html>. Accedido el 2020-06-12.
- [37] Alfresco Software Inc. Node Service. <https://docs.alfresco.com/5.2/references/dev-services-node.html>. Accedido el 2020-06-15.
- [38] Alfresco Software Inc. Content Service. <https://docs.alfresco.com/5.2/references/dev-services-content.html>. Accedido el 2020-06-15.

-
- [39] Alfresco Software Inc. FileFolder Service. <https://docs.alfresco.com/5.2/references/dev-services-filefolder.html>. Accedido el 2020-06-15.
- [40] Alfresco Software Inc. Custom form fields. https://docs.alfresco.com/process-services1.11/topics/custom_form_fields.html. Accedido el 2020-06-16.