



UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR

INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN

**Diseño y desarrollo de una aplicación web
para el etiquetado socio-semántico en el
ámbito de la gestión forestal**

Autor:

Dña. Jimena Andrade Hoz

Tutores:

**Dr. D. Guillermo Vega Gorgojo
Dr. D. Juan Ignacio Asensio Pérez**

Valladolid, 24 de septiembre de 2020

TÍTULO: **Diseño y desarrollo de una aplicación web para el etiquetado socio-semántico en el ámbito de la gestión forestal**

AUTOR: **Dña. Jimena Andrade Hoz**

TUTORES: **Dr. D. Guillermo Vega Gorgojo
Dr. D. Juan Ignacio Asensio Pérez**

DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

Tribunal

PRESIDENTE: **Eduardo Gómez Sánchez**

VOCAL: **Manuel Rodríguez Cayetano**

SECRETARIO: **Luisa M. Reguera Santos**

FECHA: **24 de septiembre de 2020**

CALIFICACIÓN:

Resumen del TFG

La generación y publicación de datos forestales en la Web es un factor clave para el correcto funcionamiento de diversas tareas relacionadas directa e indirectamente con los árboles y bosques, como son la gestión y la planificación forestal. Dos de las fuentes de datos forestales principales de España son el Inventario Forestal Nacional y el Mapa Forestal de España, que constituyen inventarios de los bosques españoles y están a disposición de la ciudadanía en forma de Datos Abiertos. Para conseguir la integración de estos conjuntos de datos en la llamada Web de Datos, se utilizan tecnologías semánticas, como es el caso del proyecto europeo *Cross-Forest*. Por otro lado, existen multitud de sitios web y aplicaciones sociales que fomentan la publicación de datos en la Web relacionados con este sector. La colaboración ciudadana permite dar un paso más allá en la obtención de datos forestales, pues permite obtener y capturar una cantidad de datos superior a la que cualquier institución podría generar por sí sola. Sin embargo, si la publicación de los datos generados por los ciudadanos no se hace con formatos y modelos de datos unificados, la reutilización de esos datos por parte de aplicaciones distintas será muy difícil, favoreciendo la creación de “silos de datos” aislados en la Web. Como solución, en este Trabajo Fin de

Grado se propone la creación de una aplicación web de carácter social que haga uso de las tecnologías de la Web Semántica, de manera que los datos publicados en ella sigan una estructura única, bien definida, que permita explotar los datos existentes en la Web de Datos. Además, dicha aplicación trata de fomentar la colaboración ciudadana, de manera que se contribuya a enriquecer la Web de Datos actual con nuevas anotaciones sociales. Esta aplicación, denominada *Timber*, permitirá a los ciudadanos recuperar y publicar información sobre los árboles de su alrededor disponibles en la Web de Datos. *Timber* pretende dar cabida a diversos tipos de usuarios (senderistas, profesionales de la gestión forestal, investigadores, estudiantes, aficionados, etc.), permitiéndoles buscar, geolocalizar, fotografiar y describir árboles de manera compartida, amigable y entretenida.

Palabras clave

Datos Abiertos Enlazados, Web Semántica, ontología, anotación social, interfaz de usuario, aplicación de una sola página.

Abstract

The generation and publication of forestry data on the Web is a key factor for the proper functioning of various tasks directly and indirectly related to trees and forests, such as forest management and planning. Two of the main sources of forest data in Spain are the National Forest Inventory and the Forest Map of Spain, which are inventories of Spanish forests and are available to the public in the form of Open Data. To achieve the integration of these data sets in the so-called Web of Data, semantic technologies are used, as is the case of the european Cross-Forest project. On the other hand, there are many websites and social applications that promote the publication of data on the Web related to this sector. Community collaboration allows to go one step further in obtaining forest data, so that it allows to obtain and capture a larger amount of data than any institution could generate on its own. However, if the publication of user-generated data is not done with unified data formats and models, the reuse of such data by different applications will be very difficult, favouring the creation of isolated 'data silos' on the Web. As a solution, this Project proposes the creation of a social web application that makes use of Semantic Web technologies, so that the data published on it follows a unique, well-defined structure that allows the exploitation of existing data on the Web of Data. Furthermore, this application tries to promote community collaboration, so that it contributes to enrich the current Web of Data with new social annotations. This application, called *Timber*, allows citizens to retrieve and publish information about the trees around them available in the Web of Data. *Timber* aims to accommodate various types of users (hikers, forest management professionals, researchers, students, amateurs, etc.), allowing them to search, geolocate, photograph and describe trees in a shared, friendly and entertaining way.

Keywords

Linked Open Data, Semantic Web, ontology, social annotation, frontend, single-page application.

Agradecimientos

Son muchas las personas a las que quiero agradecer su apoyo en esta etapa de mi vida, tanto en lo académico como en lo personal. En primer lugar, gracias Guillermo, Miguel y Asensio por confiar en mí a la hora de ofrecerme este Trabajo Fin de Grado. Gracias por guiarme durante toda la elaboración del mismo, siempre atentos para cualquier necesidad que me ha surgido. Quisiera también agradecer a Cristina Mayo, mi compañera en este proyecto, toda la ayuda que me ha ofrecido y con la que he podido aprender mucho durante todo el desarrollo del mismo. Por último, a otros participantes en las continuas reuniones para sacar el proyecto adelante, Cristobal Ordóñez y José Miguel Giménez-García.

Gracias a mis compañeros de la carrera, con los que he compartido cuatro años inolvidables. El apoyo diario que nos proporcionamos los unos a los otros hacen que el esfuerzo necesario para llegar hasta aquí haya sido más llevadero.

Finalmente, agradecer a mi familia y círculo más cercano su apoyo e impulso día a día. En especial a mis padres, por confiar siempre en mí y facilitarme el camino para llegar hasta aquí, a María, por estar ahí en los momentos más duros de este proceso y a Juan, por su apoyo incondicional.

Muchas gracias a todos.

Índice general

Índice general	VII
Índice de figuras	IX
Índice de tablas	XI
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos y alcance	4
1.3. Metodología	5
1.4. Estructura del documento	5
2. Estado del arte	7
2.1. Introducción	7
2.2. Datos abiertos	8
2.2.1. Web Semántica y Datos abiertos enlazados	10
2.2.2. Datos abiertos en el ámbito forestal	13
2.3. Anotación social	15
2.3.1. Anotación sociosemántica	16
2.4. Aplicaciones de datos abiertos en el ámbito forestal	17
2.4.1. <i>Forest Explorer</i>	17
2.4.2. <i>PlantNet</i>	18
2.5. Discusión y conclusiones	18
3. Análisis	21
3.1. Introducción	21
3.2. Visión del sistema	22
3.3. Definición de requisitos	23
3.3.1. Requisitos Funcionales	24
3.3.2. Requisitos no Funcionales	25
3.4. Casos de Uso	26
3.4.1. Gestión de usuarios	26
3.4.2. Gestión de árboles y anotaciones	29
3.4.3. Navegación y búsqueda	33
3.5. Prototipado de interfaz de usuario	35
3.6. Propuesta de curado de datos	37
3.6.1. Niveles, logros y puntos	37
3.6.2. Validación de datos	38
3.7. Conclusiones	39

4. Diseño e implementación	41
4.1. Introducción	41
4.2. Tecnologías utilizadas	42
4.2.1. Tecnologías de desarrollo software	43
4.2.2. Otras tecnologías	44
4.3. Arquitectura de la aplicación	45
4.4. Implementación	49
4.4.1. Estructura del proyecto	50
4.4.2. Detalles de implementación	51
4.5. Despliegue	53
4.6. Prototipo final y pruebas	55
4.7. Conclusiones	57
5. Conclusiones y líneas de trabajo futuro	61
5.1. Conclusiones del trabajo realizado	61
5.2. Líneas de trabajo futuro	62
Referencias	65
Referencias	65
A. Ontología de la aplicación	69
B. Glosario de la aplicación	75
C. Fichero <i>Dockerfile</i>	77
D. Manual de usuario aplicación web <i>Timber</i>	79
D.1. Introducción	79
D.2. Requerimientos	79
D.3. Instalación	79
D.4. Uso de la aplicación <i>Timber</i>	79

Índice de figuras

2.1.	Ejemplo de un grafo con triplas que describen las partes de un árbol	11
2.2.	Representación de parte de la ontología de [May20]	12
2.3.	Diagrama de la nube de <i>Linked Open Data</i>	14
3.1.	Visión de la aplicación completa	21
3.2.	Diagrama de Casos de Uso: gestión de usuarios	27
3.3.	Diagrama de Casos de Uso: gestión de árboles y anotaciones	29
3.4.	Diagrama de Casos de Uso: navegación y búsqueda	33
3.5.	Prototipo inicial de la interfaz de usuario	36
4.1.	Arquitectura cliente-servidor con API REST	45
4.2.	Arquitectura de un módulo de Angular	46
4.3.	Servicios de la arquitectura de la aplicación	48
4.4.	Resultado obtenido con el método GET al recurso Árboles	53
4.5.	Resultado obtenido con el método GET al recurso Árbol	54
4.6.	Vista de la interfaz para el caso de uso <i>DarseDeAlta</i>	56
4.7.	Vista de la interfaz para el caso de uso <i>ConsultarMisArboles</i>	57
4.8.	Vista de la interfaz para el caso de uso <i>ConsultarArbolCompleto</i>	59
4.9.	Vista de la interfaz para el caso de uso <i>RegistrarNuevoArbol</i>	60
D.1.	Vista de la página principal de <i>Timber</i>	80
D.2.	Icono de geolocalización en el mapa	80
D.3.	<i>Popup</i> con información de creador y especie	81
D.4.	Botón para acceder a la lista de árboles	81
D.5.	Botón para acceder a la información de un árbol	82
D.6.	Vista parcial del menú de navegación	82
D.7.	Menú de usuarios registrados	83

Índice de tablas

3.1. Especificación de los puntos conseguidos en función de los logros	38
3.2. Especificación de los puntos necesarios para subir de nivel	38
3.3. Especificación de los pesos para validar una anotación	39
4.1. Recursos ofrecidos por el <i>frontend</i> de <i>Timber</i>	47
4.2. Modelado de datos mediante las interfaces	49
4.3. Recursos ofrecidos por la API REST del <i>backend</i>	52

Capítulo 1

Introducción

1.1. Motivación

Las tareas llevadas a cabo en el sector forestal son muy diversas y tienen una gran importancia para obtener beneficios a nivel económico, sociocultural y ambiental. Algunas de ellas, como la gestión para la prevención y extinción de incendios o el control de plagas y enfermedades, requieren de un gran conjunto de datos con información muy variada acerca de los árboles existentes. Es por ello que la generación, gestión y almacenamiento de datos forestales es un factor clave para el correcto desarrollo de estas actividades. Además, la gestión forestal es esencial para asegurar y compatibilizar los diversos beneficios del bosque. De esta manera, los países dedican recursos para crear inventarios y mapas forestales [FG20], pues son dos de las actividades más importantes dentro del sector forestal, aunque hay muchas otras. En España existen dos proyectos mantenidos por instituciones públicas (el Ministerio de Agricultura, Pesca y Alimentación y el Ministerio para la Transición ecológica respectivamente). El primero de ellos, el Inventario Forestal Nacional (IFN), recoge información sobre “la situación, régimen de propiedad y protección, naturaleza, estado legal, probable evolución y capacidad productora de todo tipo de bienes de los montes españoles” [IFN]. El segundo, el Mapa Forestal de España (MFE), recopila la cartografía de la situación de las masas forestales españolas. Toda esta información está recogida como un conjunto de datos abierto, publicado en la Web y accesible para todos los ciudadanos. Ambos proyectos son uno de los muchos ejemplos que siguen la iniciativa de datos abiertos de España (*datos.gob.es*), denominada “Iniciativa Aporta¹”, que fue impulsada en 2009 con el fin de promocionar la apertura de la información pública y desarrollo de servicios avanzados basados en datos. Esta iniciativa sigue la filosofía del movimiento internacional denominado “*Open Data*”, que tiene por objetivo que los datos estén accesibles para cualquiera que quiera usarlos, reutilizarlos o redistribuirlos, de manera libre, accesible y sin restricciones de derechos de autor, de patentes o de otros mecanismos de control [Kit14].

La publicación de datos en la Web no solo proviene de las instituciones públicas. Las contribuciones que se hacen a ella provienen de multitud de fuentes y además, es posible encontrar en la Web información acerca de casi cualquier tema [All11]. El principal uso que se hace de estos datos es su reutilización en aplicaciones, cuyos objetivos son ayudar al desarrollo económico general y la generación de nuevos sectores y nuevos servicios para la ciudadanía. El abanico de aplicaciones que se han desarrollado reutilizando los conjuntos de datos abiertos es inmenso. Algunos ejemplos de los servicios que cubren son: servicios

¹<https://datos.gob.es/es/acerca-de-la-iniciativa-aporta>

de transporte, como es el caso de *KittyBus Madrid* o *Bus Metropolitano de Salamanca*², servicios de ecoturismo, como *Turismo y ocio Madrid* o *Castilla y León MONUMENTAL*, servicios de información meteorológica o servicios educativos, como es el caso de *Casual Learn*³ [RC20], una aplicación pensada para que alumnos de secundaria aprendan sobre historia del arte gracias a 13K actividades generadas semi-automáticamente a partir de datos de la Web.

Un tipo de aplicación que está experimentando cierto auge son las aplicaciones de carácter social. Las aplicaciones sociales toman relevancia en esta continua generación de datos, pues favorecen una mejor comunicación y colaboración entre los ciudadanos, una gestión más eficaz del conocimiento y una innovación más rápida en muchos campos de la ciencia [Yu07]. Dentro del ámbito de las aplicaciones sociales, destacan aquellas que dan apoyo a la denominada “ciencia ciudadana”. La ciencia ciudadana [Bre09] puede describirse como el compromiso del público general en actividades de investigación científica; es decir, los ciudadanos contribuyen activamente a la ciencia con su esfuerzo intelectual o dando soporte al conocimiento con sus herramientas o recursos. Los participantes proveen datos experimentales o equipos a los investigadores. Los voluntarios, a la vez que aportan valor a la investigación, adquieren nuevos conocimientos o habilidades, y un mejor conocimiento del método científico de una manera atractiva. Como resultado de este escenario abierto, colaborativo y transversal, las interacciones entre ciencia-sociedad-políticas investigadoras mejoran, conduciendo a una “investigación más democrática”, basada en la toma de decisiones provenientes de evidencias informadas por parte de científicos amateur o no profesionales [Pro13]. Un buen ejemplo de este tipo de aplicación es *Waze*⁴, una de las aplicaciones de tráfico y de navegación abierta y distribuida en tiempo real más popular en todo el mundo. *Waze* aprovecha la colaboración de los usuarios en tiempo real para mejorar la congestión y tomar las mejores decisiones de planificación urbana. Además, los gobiernos locales proporcionan a *Waze* datos de construcción, cierre de choques y cierre de carreteras en tiempo real a través de plataformas de datos abiertos [Waz17].

La ciencia ciudadana trae consigo una mejora significativa en el crecimiento de diferentes campos del conocimiento [Bre09], como es el caso del sector forestal. Las herramientas de carácter social facilitan el desarrollo de diversas tareas relacionadas con este sector, tales como la recogida de datos en el campo o la anotación de árboles, donde los ciudadanos pueden subir y compartir información de los árboles de su alrededor. Un ejemplo es *Curio*⁵, una aplicación que utiliza el intercambio de información entre sus usuarios y el mapeo comunitario para explorar y aprender sobre el arbolado urbano y compartir los descubrimientos entre ellos. Este tipo de aplicaciones consiguen obtener mayor cantidad de datos de a los que puede llegar cualquier institución, además de poder utilizarlos como aprendizaje a nivel escolar, universitario o incluso aficionado. Hoy en día existen multitud de sitios web y aplicaciones con información relacionada con el sector forestal (*PlantNet*⁶, *Curio*, *TreeTalk*⁷, etc.) pero, ¿podemos fiarnos de los datos publicados?

²<https://www.bmsalamanca.com/es/inicio>

³Esta aplicación ha sido desarrollada por el grupo de investigación GSIC-EMIC (Grupo de Sistemas Inteligentes y Cooperativos - Educación, Medios, Informática y Cultura), dentro del cual se ha desarrollado este Trabajo Fin de Grado

⁴<https://www.waze.com/es>

⁵<https://www.curio.xyz/about>

⁶<https://plantnet.org/en/>

⁷<https://www.treetalk.co.uk/>

La anotación social en la Web trae consigo algunos problemas. Uno de ellos es la falta de control sobre el contenido publicado y la posible imprecisión de los datos generados [Lyt09]. Esta problemática hace que las aplicaciones de carácter social necesiten de determinados sistemas de control (expertos en el tema o técnicas automatizadas) para dar como verdaderos los contenidos que las mismas albergan. Por otro lado, la información publicada no suele seguir ningún tipo de estructura, pues resulta difícil utilizar modelos y vocabularios comunes de manera consistente con comunidades grandes y heterogéneas. Esto hace que los datos se encuentren dispersos y sin relación explícita entre ellos [Bre09], provocando que, en muchas ocasiones, una simple búsqueda se convierta en una tarea compleja. La principal causa de ello es que la Web está construida de tal manera que sus documentos sólo contienen información suficiente para que los dispositivos los presenten, no para que los entiendan [Yu07], favoreciendo su presentación pero no el procesamiento de su contenido. Un tercer problema, centrado en el carácter social de la web (“Web Social”), es que los sitios web sociales están aislados unos de otros. Por ejemplo, diferentes discusiones en línea pueden contener conocimientos y temas complementarios, partes segmentadas de una respuesta que una persona puede estar buscando, pero las personas que participan en una discusión no tienen fácil acceso a la información sobre discusiones relacionadas en otros lugares. La Web Social crea un conjunto de “silos de datos” únicos, es decir, hay muchos sitios, comunidades y servicios que no pueden interoperar entre sí, en los que la reutilización y la interconexión de los datos es difícil y engorrosa [Bre09].

Como solución al segundo y tercer problema mencionado, y a otros que se detallarán en el siguiente capítulo, se propone la **Web Semántica**. Ésta no pretende sustituir a la Web actual, sino ser una extensión de la misma, en la que se dé un significado bien definido a la información, y permita a los ordenadores y a las personas trabajar en cooperación [BL01]. Con este nuevo enfoque de estructuración de los datos se busca dejar a un lado la inconsistencia, falta de conexión de los mismos, para dar paso a la integración, conexión y reutilización de ellos. Así, las aplicaciones pueden utilizar estas conexiones y organizar sus datos de manera que proporcionen una descripción integrada de los objetos del mundo y sus relaciones en lugar de su presentación [All11]. A éstas se las denomina “aplicaciones semánticas” (*semantic applications*). Los datos que manejan siguen unos determinados modelos estructurados gracias al uso de ontologías, que facilitan la creación de vínculos con otros dentro de la Web, lo que se denomina formalmente “*Linked Open Data*” (datos abiertos enlazados). Los enlaces que se generan forman parte de la conocida “Web de Datos”. Por otro lado, surge el concepto de “**Web Semántica Social**” (*Social Semantic Web*), una combinación de la Web Social y la Web Semántica donde los datos sociales generados puedan interconectarse mediante tecnologías semánticas. Con este nuevo enfoque, las aplicaciones semánticas son mejoradas gracias a la riqueza de conocimiento inherente al contenido generado por los usuarios, dando pie a lo que se denominan “aplicaciones socio-semánticas” [Bre09]. Dichas aplicaciones utilizan la anotación semántica, que se refiere al proceso mediante el cual se incluyen metadatos semánticos en los recursos web. Una anotación semántica combina conceptos de metadatos y ontologías, es decir, los campos de los metadatos son asociados con términos en una ontología. De este modo, las anotaciones socio-semánticas son anotaciones que se realizan de manera manual, generalmente por usuarios humanos. Son las aplicaciones internamente las encargadas de asociar los datos de las anotaciones con los términos de las ontologías.

Tras conocer la inexistencia de aplicaciones socio-semánticas en el sector forestal, se

plantea en el presente Trabajo Fin de Grado la creación de una aplicación web de carácter social y semántico para la anotación de árboles, con la que se puedan solventar los problemas mencionados anteriormente. Dicha aplicación utilizará una ontología para permitir realizar anotaciones semánticas a la comunidad de usuarios, de manera que los datos generados en ella seguirán una estructura bien definida que permitirá su reutilización en otras aplicaciones. La interfaz de usuario tomará un papel relevante en este proceso, pues deberá encargarse de la correcta adaptación de los datos introducidos por los usuarios según la ontología. Además, con el enfoque social se pretende fomentar la colaboración de los ciudadanos y aprovecharse de la ciencia ciudadana para obtener mayor cantidad de datos relacionados con los árboles, permitiendo enriquecer el conjunto de datos existente con anotaciones de los usuarios en zonas a las que no llega el IFN, por ejemplo jardines.

1.2. Objetivos y alcance

El objetivo del presente Trabajo Fin de Grado (TFG) es diseñar y desarrollar una **interfaz de usuario de una aplicación web adaptable⁸ para la anotación socio-semántica de árboles** que permita a diversos tipos de usuarios (senderistas, profesionales de la gestión forestal, investigadores, estudiantes, etc.) buscar, geolocalizar, fotografiar y describir árboles de manera compartida. La aplicación web desarrollada permitirá generar, de manera paulatina, un conjunto de datos abierto complementario a los ya mantenidos por las instituciones públicas, como es el caso del Inventario Forestal Nacional. Además, el conjunto de datos generado hará uso de tecnologías de la Web Semántica para poder enriquecerlos mediante enlaces con los ya disponibles en la Web de Datos. Esta aproximación al uso de tecnologías de la Web Semántica en el ámbito de la gestión forestal ya está siendo explorada por el grupo de investigación GSIC-EMIC (Grupo de Sistemas Inteligentes y Cooperativos - Educación, Medios, Informática y Cultura), al que pertenecen los tutores del presente Trabajo Fin de Grado. Un resultado reciente es el *Explorador Forestal*⁹, desarrollado dentro del proyecto europeo *Cross-forest*¹⁰. Del mismo modo, en este Trabajo Fin de Grado se proponen y llevan a cabo pruebas de evaluación tecnológicas y con usuarios reales del prototipo desarrollado.

El desarrollo de la interfaz de usuario descrita se ha llevado a cabo paralelamente con Cristina Mayo, alumna del Máster en Ingeniería de Telecomunicación, quien se encarga del desarrollo de los componentes servidores necesarios (*backend*) y la creación de la ontología utilizada para la integración de los datos creados con la infraestructura de datos del *Explorador Forestal* y, consecuentemente, con el resto de la Web de Datos. Este *backend* facilitará una API REST (*Application Programming Interface REpresentational State Transfer*) que proporcionará los servicios necesarios [Fie00] para dotar de funcionalidad al *frontend* de este TFG. Así pues, el objetivo principal conjunto es la creación de un prototipo de aplicación web para la anotación socio-semántica de árboles que persigue tres metas fundamentales. En primer lugar, apoyarse en la Web Semántica como mecanismo para la creación e integración de datos forestales en la Web, permitiendo su futura reutilización en otras aplicaciones. En segundo lugar, aprovechar los recursos forestales y las bases de datos institucionales ya existentes como elemento de partida en esta iniciativa;

⁸En inglés el término se traduce como *responsive*, y hace referencia a una técnica de diseño web que busca la correcta visualización de una misma página web en distintos dispositivos

⁹<https://forestexplorer.gsic.uva.es/>

¹⁰<https://crossforest.eu/>

favoreciendo la reutilización de datos, una de las características más relevantes de la Web Semántica. Finalmente, y en relación con el carácter social de la aplicación, se pretende dar visibilidad a este campo, fomentar su divulgación de manera adecuada y contribuir a la generación de datos de manera colaborativa.

1.3. Metodología

Para completar los diferentes objetivos del presente TFG se ha seguido una metodología iterativa e incremental, propia del Proceso Unificado de Desarrollo de Software [Jac00]. Este marco de trabajo propone múltiples ciclos, cada uno comprendiendo las fases de Arranque o Inicio, Elaboración, Construcción y Transición. En este marco de desarrollo, cada una de las fases anteriores comprende a su vez una serie de iteraciones que ofrecen como resultado un incremento de la aplicación desarrollada, que añade o mejora las funcionalidades del sistema en desarrollo. La selección de qué funcionalidad se desarrolla en cada iteración está impulsada por la mitigación de los riesgos del proyecto. Con cada iteración, gradualmente se refinan las decisiones estratégicas y tácticas, convergiendo finalmente en una solución que cumple con los requisitos reales (y que no estaban previamente declarados) del usuario final [Boo07].

Los ciclos del proyecto se fueron centrando en diferentes fases según se fue avanzando en el desarrollo de la aplicación. A continuación, se especifican las funcionalidades en las que se centró cada una de las iteraciones:

- Las primeras iteraciones del proyecto se centraron en la fase de análisis, en la que tras varias semanas de reuniones con los tutores del TFG y expertos en el campo de la ingeniería forestal, se analizaron y discutieron los requisitos y funcionalidades que la aplicación debería cumplir.
- En las siguientes iteraciones, el peso recayó en gran medida en el diseño de un prototipo para la interfaz y la discusión de las tecnologías que se iban a utilizar, siguiendo además con la creación de las especificaciones más elaboradas de los requisitos de la aplicación.
- Las últimas iteraciones han consistido en la implementación propia de la aplicación y las posteriores pruebas de la misma.

1.4. Estructura del documento

A continuación se describen las partes de las que consta el documento, que se encuentra dividido en cinco capítulos, entre los que se incluye el capítulo presente. En el capítulo 2 se indaga en los Datos Abiertos Enlazados y su intervención en la ingeniería forestal, además de la existencia de aplicaciones sociales y la importancia de la ciencia ciudadana para el crecimiento de este sector. En el capítulo 3 se describe todo el proceso de análisis de la aplicación. Incluyendo los requisitos básicos que se quieren alcanzar, un prototipo de interfaz de usuario y una propuesta para el curado de los datos publicados. El capítulo 4 incluye la descripción de las tecnologías utilizadas en el diseño de la interfaz de usuario, así como su arquitectura y algunos detalles de implementación. En el capítulo 5 se encuentran las conclusiones sobre los resultados obtenidos, así como indicaciones sobre los pasos futuros que habría que dar para continuar con el desarrollo de la aplicación. Por

último, se recoge un apéndice con la ontología diseñada para la aplicación, un glosario con la terminología básica de la misma, otro con instrucciones técnicas para el despliegue de la aplicación en producción y, finalmente, uno que describe un breve manual de usuario de la aplicación creada.

Capítulo 2

Estado del arte

2.1. Introducción

La Web Semántica propone mitigar problemas provenientes de la Web, como la búsqueda de contenidos o la reutilización e integración de datos, que se deben a diversas razones como la orientación a documentos de la Web clásica, la falta de estructuración de los datos (y con modelos comunes) y la falta de entrelazado entre datos provenientes de diferentes fuentes. Además, la Web Semántica está ligada al movimiento de Datos Abiertos Enlazados, que persigue la generación de datos que estén conectados entre sí, disponibles para todos los ciudadanos y reutilizables para diferentes fines y aplicaciones [Biz11]. En este capítulo se da una visión de lo que estos conceptos significan y las ventajas que pueden obtenerse. Más en concreto, cómo utilizar estos beneficios dentro del ámbito forestal, con el objetivo último de identificar los requisitos que ha de satisfacer la aplicación propuesta y desarrollada en este Trabajo Fin de Grado. Además, se introduce la Web Social, una manera distinta de utilizar la Web, sobre todo en la generación de contenidos gracias a la participación activa de los ciudadanos, que favorece en gran medida la publicación de datos en la Web [Bre09]. La colaboración ciudadana ha contribuido al crecimiento de múltiples campos del conocimiento, como se verá en el ámbito forestal. Una vez más, se recurre a las tecnologías semánticas para solventar problemas derivados de la Web Social.

Este capítulo se estructura como sigue: la sección 2.2 pretende dar una visión general de lo que son los datos abiertos y lo que estos suponen para diferentes actores como ciudadanos en general, empresas o el propio gobierno. Esta sección se divide a su vez en dos, la 2.2.1 donde se habla de los Datos Abiertos Enlazados, la mejor manera de publicar datos abiertos en la Web y la subsección 2.2.2, que trata el caso general de los Datos Abiertos Enlazados dentro del ámbito forestal. La sección 2.3 explica los fundamentos de la anotación social, así como sus debilidades. La manera de solucionar los problemas que trae consigo la anotación social se trata en la subsección 2.3.1. La sección 2.4 centra sus objetivos en descubrir algunas aplicaciones sociales impulsadas por la colaboración ciudadana y la generación y reutilización de Datos Abiertos forestales. Finalmente, en la sección 2.5, se expone una visión global de todo el capítulo, proponiendo una aplicación web social basada en tecnologías de la Web Semántica.

2.2. Datos abiertos

El acceso a conjuntos de datos ha sido, en general, limitado debido al gasto que supone generarlos y los recursos necesarios para producirlos y mantenerlos. Incluso cuando éstos han estado relativamente disponibles, han requerido de equipos e instrumentos especializados y conocimientos técnicos y contextuales relativos a un campo o tema, gran parte de los cuales están más allá de las capacidades de la población en general. En consecuencia, los datos, la información y los conocimientos derivados de ellos han estado tradicionalmente cerrados por naturaleza, encerrados en una institución o un archivo [Kit14].

El movimiento de **datos abiertos** (*open data movement*) [Kit14] busca transformar radicalmente esta situación, tanto abriendo los datos para una mayor reutilización, como proporcionando herramientas de investigación fáciles de usar que eliminen la necesidad de habilidades especiales para poder utilizarlos. Los datos abiertos se basan en tres principios básicos que son: apertura, participación pública y colaboración ciudadana. Promueve que los conjuntos de datos sigan unos patrones estandarizados, de forma que puedan ser fácilmente accesibles tanto para los ciudadanos como para las máquinas. De esta manera, los datos pueden exportarse automáticamente e integrarse en otros productos de información para crear servicios nuevos. Uno de sus objetivos es democratizar la capacidad de producir información y conocimientos, en lugar de limitar el poder de los datos a sus productores y a quienes pueden pagar por su acceso. Para conseguirlo, los datos abiertos deben ir acompañados de mecanismos de descubrimiento de datos, para emitir indicaciones de calidad y fiabilidad de estos, además de utilizar formatos y normas de datos que mejoren la interoperabilidad. También se deben proporcionar infraestructuras de datos fáciles de utilizar que faciliten su publicación periódica, la generación de nuevos conjuntos y el acceso a los mismos, así como conjuntos de instrumentos básicos y especializados que mejoren su utilización y su análisis [Ope18].

Tras el proceso de publicación de los datos es importante una buena gestión de los mismos para permitir su posterior integración y reutilización tanto de los datos como de los conocimientos inherentes a ellos. Este proceso es altamente necesario en datos académicos y científicos, ya que la infraestructura actual impide extraer el máximo beneficio de las inversiones realizadas. Para solventar esta situación surge lo que se denominan los Principios de Datos Equitativos, más comúnmente conocidos en inglés como *FAIR Data Principles* [Wil16]. Esta iniciativa está ligada al movimiento de datos abiertos y hace hincapié en que los datos sean localizables, accesibles, interoperables y reutilizables (*Findability, Accessibility, Interoperability, and Reusability*). Estos principios se centran específicamente en mejorar la capacidad de las máquinas para encontrar, acceder, interoperar y reutilizar datos sin intervención humana o con una intervención mínima. Garantizar la interoperabilidad permite la unión de diferentes conjuntos de datos, dando paso a la creación de nuevos conjuntos más complejos y revelando nuevos conocimientos. Gran cantidad de interesados salen beneficiados con este proceso: investigadores que deseen compartir, reutilizar y obtener crédito con los datos, editores profesionales de datos, creadores de programas informáticos y una gran comunidad de ciencias de la información que extrae, integra y analiza tanto datos nuevos como existentes para avanzar en la investigación y el desarrollo [Wil16]. Existen muchos repositorios que ya hacen uso de los principios *FAIR*, como es el caso de *ISA commons*¹, una comunidad en crecimiento que facilita la recopila-

¹<https://www.isacommons.org/>

ción, la conservación, la gestión y la reutilización de conjuntos de datos científicos. Los dos últimos principios se consiguen gracias a la publicación de los datos como datos enlazados (*linked data*), concepto que se explica en la siguiente sección.

Desde 2010, muchos países y organizaciones internacionales se han unido al movimiento de datos abiertos, haciendo que miles de conjuntos de datos que anteriormente estaban restringidos sean ahora de naturaleza abierta y se puedan utilizar libremente². Por ejemplo, una mujer en Dinamarca desarrolló *findtoilet.dk* que muestra todos los baños públicos daneses, para que cualquiera pueda aprovecharse de esta información. En los Países Bajos está disponible un servicio, *vervuilingsalarm.nl*, que te avisa con un mensaje si la calidad del aire de tu vecindario alcanzará, al día siguiente, el umbral previamente definido. Todos estos ejemplos de servicios usan datos abiertos [Ope18].

En España se están llevando a cabo proyectos que pretenden colocar al país en la vanguardia de la reutilización de la información del sector público en Europa, con la plataforma *Datos.gob.es*, el Catálogo de Información Pública del Sector Público³. Este proyecto está enmarcado dentro de la *Iniciativa Aporta*⁴, puesta en marcha en 2009 con el fin de promocionar la apertura de la información pública y desarrollo de servicios avanzados basados en datos y cuyo fin es promover nuevos productos y servicios por parte del sector privado y la sociedad civil en beneficio de la sociedad.

La apertura de datos resulta beneficiosa en múltiples sectores, como por ejemplo el sector académico o el sector económico, donde nuevos productos y compañías están reutilizando datos públicos para mejorar sus servicios. Existen también diferentes grupos de personas y organizaciones que pueden beneficiarse de la disponibilidad de datos abiertos, incluido el mismo gobierno. A primera vista, parece haber pocas desventajas en la apertura de datos. Sin embargo, esta iniciativa no ha sido universalmente acogida. De hecho, están surgiendo críticas que sostienen que este movimiento no es política ni económicamente beneficioso [Kit14]. Podemos dividir estas críticas en 3 grupos. La primera de ellas (i) argumenta que estas iniciativas no parten de un terreno neutral, sino que están promovidas por la ideología política y económica, donde los partidos políticos y las empresas se han apropiado del movimiento de datos abiertos en su beneficio [Bat12]. La segunda de ellas (ii) tiene relación con el acceso a los datos y su utilización de manera desigualitaria. Según [Joh14], existe un alto grado de privilegio social en los datos del sector público con respecto a qué datos se generan, en relación con quién y para qué. Esto puede provocar la propagación de las injusticias sociales y reforzar los intereses de los que dominan. La tercera (iii) se centra en la sostenibilidad, utilidad y facilidad de uso. El movimiento *open* se ha centrado, en muchas ocasiones, solo en crear iniciativas de datos abiertos, dejando a un lado la atención a la sostenibilidad y forma de utilización de los mismos. Se ha visto en un estudio [Hel12] que, como resultado, se han creado multitud de sitios de datos abiertos que funcionan más bien como depósitos o “vertederos” de datos que como lugares donde reutilizar e interoperar con ellos.

Ahora bien, estas críticas no promueven que se abandone la tendencia a la apertura de datos, sino que las iniciativas *open* tengan más en cuenta la forma en que se ponen a disposición los datos, cómo se utilizan y cómo se financian. En relación a esto, podemos ver

²<https://dataremixed.com/2013/08/worldwide-open-data-sites/>

³<https://datos.gob.es/>

⁴<https://datos.gob.es/es/acerca-de-la-iniciativa-aporta>

un ejemplo de estas críticas y la manera de solventarse que se propone en este TFG, pues uno de los objetivos de la aplicación propuesta es que ciudadanos no expertos en datos sean capaces de aprovechar los datos abiertos existentes en el sector forestal, revirtiendo el aprovechamiento de los datos abiertos de manera exclusiva por las grandes corporaciones.

La forma en la que se publican los datos toma relevancia en este momento, y se verá a continuación que la publicación de datos abiertos enlazados es la mejor manera de publicar datos abiertos en la Web.

2.2.1. Web Semántica y Datos abiertos enlazados

La Web Semántica se basa en la infraestructura de la Web (HTTP [Fie99] y URIs [BL05]) y se puede considerar como la evolución natural de la Web clásica en la que los datos presentados poseen un formato único y procesable por las máquinas. Ésta no pretende sustituir a la Web actual, sino que Tim Berners-Lee la define en [BL01] como: “La Web Semántica no es una Web independiente sino una extensión de la actual, en la que se da un significado bien definido a la información, lo que permite a las computadoras y a las personas trabajar en cooperación”. Mientras que las unidades primarias de la Web de hipertexto son los documentos HTML (*HyperText Markup Language*) conectados por hipervínculos, los datos de la Web Semántica se basan en documentos que contienen datos en formato RDF⁵ (Marco de Descripción de Recursos, *Resource Description Framework*). En ella, se habla de “recursos” para referirse a cualquier cosa del mundo.

La Web Semántica utiliza **ontologías** para crear vocabularios de conceptos y las relaciones entre ellos dentro de las diferentes áreas de conocimiento. Una ontología es una formalización de un dominio y se utiliza para definir el vocabulario del mismo [All11]. Los dominios pueden ser de cualquier tema, como por ejemplo la medicina, los transportes, la comida o las plantas. Las ontologías facilitan el intercambio de información y conocimientos. En ocasiones se habla también de “vocabulario” como sinónimo de ontología. Existen muchas ontologías disponibles para todo tipo de dominios (*Linked Open Vocabularies*⁶, LOV). La adopción de ontologías comunes por parte de todos los participantes en la Web Semántica es clave, ya que de esta manera se puede trabajar de forma autónoma pero con la garantía de “hablar el mismo idioma”. Además, cualquiera puede proponer su propia ontología. Esta opción es típica si no hay un vocabulario del dominio que se quiere anotar. Actualmente existe un gran número ontologías disponibles, pero en la mayoría de los casos hay que hacer ajustes, extensiones o alineamiento de las mismas, requiriendo la creación de nuevos vocabularios. Este es el caso de la ontología que se ha creado en [May20] para describir los datos de la aplicación web propuesta en este TFG. La ontología completa puede consultarse en el Apéndice A.

Las tecnologías fundamentales de la Web Semántica están estandarizadas por la W3C (*World Wide Web Consortium*) y se describen brevemente a continuación:

- **RDF:** el modelo RDF codifica los datos en forma de “triplas”, formadas por: sujeto (recurso que se describe), predicado (propiedad del recurso) y objeto (valor de la descripción). Por ejemplo «Un árbol tiene tronco» sería una tripla donde “Árbol” es el sujeto, “tiene” el predicado y “tronco” el objeto. Un conjunto de triplas

⁵<https://www.w3.org/TR/rdf11-concepts/>

⁶<https://lov.linkeddata.es/dataset/lov>

puede representarse como un grafo acíclico dirigido (un tipo de grafo) y representa una vista estructural de los nodos y los arcos (enlaces) de datos RDF [All11]. Se puede ver un ejemplo en la Figura 2.1, donde los sujetos y los objetos son los nodos, y los predicados la descripción de los arcos. Además, RDF aplica el concepto de URI para fusionar grafos. Un nodo de un grafo se fusiona con un nodo de otro si tienen la misma URI. De esta manera se pueden enlazar datos que provienen de diferentes conjuntos de la Web.

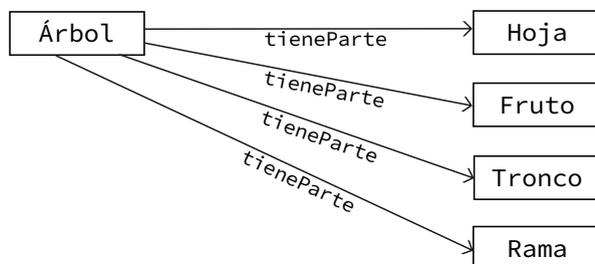


Figura 2.1: Ejemplo de un grafo con triplas que describen las partes de un árbol

- **RDFS** (*RDF Schema*⁷): RDFS permite definir las primitivas necesarias para describir las clases, instancias y relaciones de los datos representados en RDF. Es una extensión semántica del RDF [Bre09]. En la Figura 2.2 se muestra un ejemplo de la representación de una ontología en RDFS.
- **OWL** (*Web Ontology Language*⁸): el lenguaje de ontología web, es un lenguaje diseñado con el fin de representar el conocimiento de una manera más rica y compleja que RDFS. OWL extiende la noción de clases y propiedades definidas en RDFS y proporciona nuevos axiomas para definir características y restricciones avanzadas en relación con las clases y propiedades [Bre09].
- **SPARQL** (*Protocol and RDF Query Language*): es un lenguaje para consultar datos RDF. SPARQL es esencialmente un lenguaje declarativo basado en la coincidencia de patrones de grafos.

Como se ha mencionado anteriormente, en [May20] se ha diseñado una ontología para el dominio de anotación de árboles de la aplicación propuesta en este TFG. Se puede consultar su contenido completo en el Apéndice A, donde se especifican tanto las clases como las propiedades que la componen. La URI de la ontología es <http://timber.gsic.uva.es/sta/ontology/>, aunque a partir de ahora, se utilizará el prefijo *sta* (*Social Tree App*) para acortar los nombres de los términos definidos en la ella. Como ejemplo de su utilización, en la Figura 2.2 se representa la parte de la ontología destinada a la anotación de un árbol. Un árbol (clase *Tree*) puede tener anotaciones de tipo posición, especie e imagen (clases *PositionAnnotation*, *SpeciesAnnotation* y *ImageAnnotation* respectivamente).

⁷<https://www.w3.org/TR/rdf-schema/>

⁸<https://www.w3.org/TR/owl-features/>

Todas ellas son especializaciones de la clase `Annotation`. Además, y como se explicará en los capítulos siguientes, los árboles anotados incluirán anotaciones primarias (clase `PrimaryAnnotation`). Los arcos especifican las relaciones que existen entre cada una de las clases especificadas.

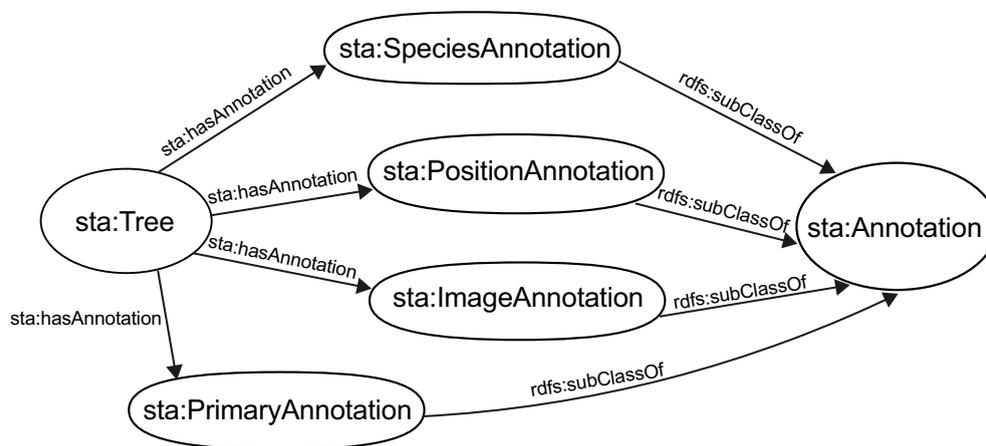


Figura 2.2: Representación de parte de la ontología de [May20]

La Web Semántica no sólo consiste en exponer datos en la Web, sino en crear enlaces para que cualquier persona o máquina pueda explorar la Web de Datos. El término de “**Datos Enlazados**” (*Linked Data*, LD) hace referencia a un conjunto de buenas prácticas para publicar y conectar datos estructurados en la Web. Se basa en tecnologías de la Web clásica pero extendiendo su uso de manera que la información compartida pueda ser más fácilmente procesable por las máquinas. Los datos enlazados constituyen un método para exponer, compartir y conectar datos en la Web Semántica [Biz11]. Cuando se tienen datos enlazados, se pueden encontrar fácilmente otros relacionados. Los cuatro principios de LD [BL06] son:

1. Usar URIs para identificar los recursos publicados en la Web
2. Utilizar URIs HTTP para localizar y consultar estos recursos
3. Cuando se acceda a una URI, se proporcione información útil, usando los estándares (RDF y SPARQL)
4. Incluir enlaces a otras URIs relacionadas con los datos contenidos en el recurso, de forma que se favorezca el descubrimiento de información en la Web

Los datos enlazados se propusieron para revitalizar la Web Semántica, puesto que estaba fracasando: mucho esfuerzo en la propuesta de lenguajes de ontologías, pero muy pocos resultados tangibles. Se podría decir que supuso un cambio notable, sobre todo a raíz de los datos abiertos enlazados, aunque la Web Semántica sigue siendo difícil de adoptar, limitándose a grandes corporaciones y aplicaciones científicas (biomedicina, lingüística...)

principalmente.

Ahora bien, hay que diferenciar entre datos abiertos y **datos abiertos enlazados** (*Linked Open Data*, LOD). Tim Berners-Lee lo explica de manera sencilla: los datos abiertos enlazados son datos enlazados que están publicados en la Web bajo una licencia abierta, lo que permite su reutilización gratuita [BL06]. Uno de los objetivos de la W3C es ampliar la Web actual con conjuntos de datos en RDF y que sean a su vez de carácter abierto, ya que la mayoría de datos abiertos publicados en la Web no están enlazados. De esta manera se solventa la tercera crítica comentada en el apartado anterior, pues se evitaría la creación de silos y depósitos de datos abiertos que no se pueden reutilizar. Es por ello que los datos abiertos enlazados son la mejor manera de publicar datos abiertos en la Web.

Existe una clasificación de los datos propuesta por Tim Berners-Lee basada en la obtención de estrellas. Este esquema de estrellas se creó para incentivar a los proveedores de datos a hacerlo lo mejor posible, con el objetivo de conseguir todas ellas. La clasificación comienza con una estrella y los datos van obteniendo estrellas cuando se eliminan los formatos propietarios y se añaden enlaces. La primera se consigue cuando los datos publicados en la web tienen licencia abierta, es decir, son datos abiertos. La segunda estrella se obtiene si los datos abiertos están disponibles como datos estructurados, capaces de ser procesados por máquinas (una hoja de Excell por ejemplo). La tercera se otorga a aquellos datos abiertos disponibles en formatos abiertos. La cuarta estrella se consigue cuando estos datos utilizan estándares abiertos del W3C (RDF y SPARQL). La quinta y última se obtiene cuando los datos publicados, además de cumplir todos los requisitos anteriores, se enlazan con otros datos de la Web de Datos. El resultado más relevante del LOD es la “Nube de Datos Abiertos Enlazados”, más conocida como *The Linked Open Data Cloud*⁹, el conjunto de las principales fuentes de datos publicadas siguiendo los principios de *Linked Data* y con una licencia de datos abierta. En la Figura 2.3 se muestra el conjunto de datos actual, que contiene 1260 conjuntos de datos con 16187 enlaces (a mayo de 2020).

2.2.2. Datos abiertos en el ámbito forestal

Los datos abiertos forestales pueden ayudar a mejorar la gestión y la vigilancia de los bosques, además de impulsar la ciencia y la investigación de los mismos. Es por ello que se está realizando un esfuerzo importante en la generación y publicación de datos abiertos forestales, generándose cada vez más datos. El *big data* forestal implica que cada vez existen más datos forestales grandes (como datos genéticos y LIDAR), veloces [Lan01] (como el Internet de las Cosas utilizando sensores de humedad y temperatura en árboles) y variados (inventarios, cobertura arbórea, suelos, incendios, especies, etc.), siendo esta última dimensión la relevante en el TFG. Existen varias iniciativas de datos abiertos dentro del sector forestal a nivel mundial, como por ejemplo *Trees4Future*¹⁰, un proyecto integrador de Infraestructura de Investigación Europea que tuvo como objetivo integrar, desarrollar y mejorar las principales infraestructuras de investigación en genética forestal y silvicultura. Todos los datos generados eran guardados en formato RDF siguiendo una ontología forestal que los permitía enlazarse con otros datos de la Web Semántica [Lok14]. Este proyecto terminó en 2016 y, aunque no consiguió grandes avances en lo que respecta a LOD, es un claro ejemplo de este tipo de iniciativas en el sector forestal.

⁹<https://lod-cloud.net/>

¹⁰<http://www.trees4future.eu/>

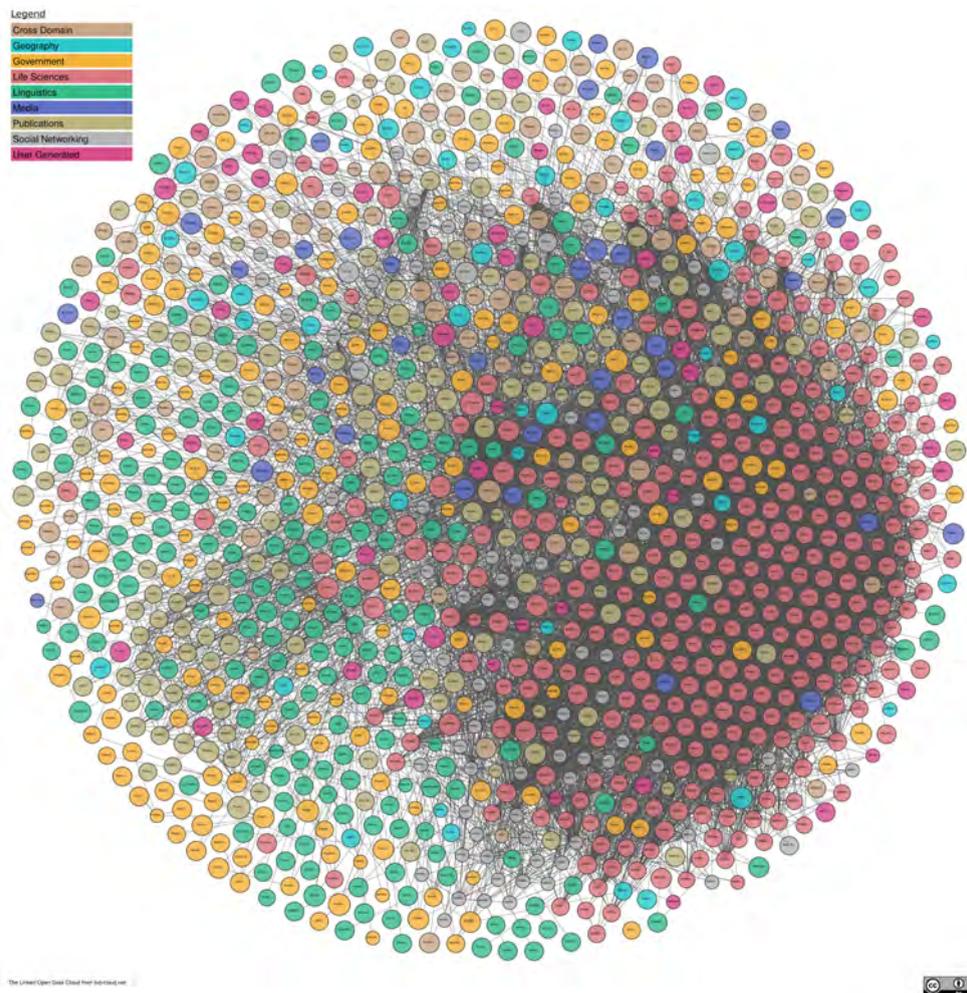


Figura 2.3: Diagrama de la nube de *Linked Open Data*

En el caso concreto de España, el Ministerio de Agricultura, Pesca y Alimentación realiza un inventario continuo, en el que se repiten las mediciones cada 10 años, que da como resultado el Inventario Forestal Nacional (IFN) y el Mapa Forestal de España (MFE). Ambos están encaminados a obtener la máxima información posible sobre la situación, régimen de propiedad y protección, naturaleza, estado legal, probable evolución y capacidad productora de todo tipo de bienes de los montes españoles. Los datos obtenidos son publicados en la Web como un conjunto de datos abiertos [IFN]. Sin embargo, estos datos están divididos en varios archivos publicados en una base de datos SQL y en formato propietario que requiere una licencia Microsoft. Por lo que los resultados recogidos resultan, difícilmente accesibles y en formatos que no están disponibles para el público en general. Se podría decir que estos datos tan solo obtienen dos de las cinco estrellas del LOD.

A raíz de la publicación de estos datos, surge el proyecto *Cross-Forest*¹¹. Una de sus actividades principales es la transformación y publicación de los datos provenientes del IFN y el MFE, generando ontologías de datos forestales accesibles para el sector público [GG19]. Está liderado por investigadores del grupo de investigación GSIC/EMIC (Grupo

¹¹<https://crossforest.eu/>

de Sistemas Inteligentes y Cooperativos / Educación, Medios, Informática y Cultura) de la Universidad de Valladolid y el iuFOR (Instituto Universitario de Investigación en Gestión Forestal Sostenible). La Web Semántica toma en este proyecto especial relevancia, pues permite la publicación de datos de una manera autodescriptiva e interoperable. Es por ello que para el proyecto (aún en desarrollo) se han diseñado un total de cinco ontologías centrales, que permitirán representar todos los datos provenientes del IFN y del MFE, así como posiciones y mediciones de los mismos. Además, están interrelacionadas entre sí (son datos enlazados) y enriquecidas con otras ontologías externas existentes [FG20]. Los datos RDF obtenidos están publicados en un almacén de triplas, y se puede acceder a ellos y hacer consultas a través de la URL del punto SPARQL¹².

2.3. Anotación social

La Web tradicional ha ido evolucionando desde sus orígenes para satisfacer las diferentes necesidades de los ciudadanos. El papel del usuario en la Web también ha cambiado, pasando de ser tan solo un consumidor de contenido a un participante activo en su creación. Desde principios del 2000, se empezó a observar una fuerte tendencia a desarrollar tecnologías que facilitasen la colaboración y participación de los usuarios en los sitios web. A este aspecto de colaboración e intercambio se le denomina “**Web Social**” y se utiliza para describir un subconjunto de interacciones entre usuarios en la Web que son altamente sociales, conversacionales y participativas [Bre09].

La Web Social describe una Web en la que los usuarios pueden reunirse, colaborar y compartir contenidos en espacios sociales mediante elementos etiquetados, flujos de actividad, funcionalidad de redes sociales, etc. Además, permite la participación de los usuarios en cualquier tema mediante blogs y etiquetado, lo que ha demostrado que la “sabiduría de las multitudes” puede generar un conocimiento mucho más valioso que el que se crea con individuos separados [Bre09]. Un claro ejemplo es la *Wikipedia*¹³, la enciclopedia de contenido libre que todos pueden editar y que se ha convertido en la más grande del mundo. A la creación de inteligencia colectiva se le conoce también como “**ciencia ciudadana**” [Lyt09].

Los sitios web sociales proporcionan acceso al contenido aportado por la comunidad. Este contenido puede abarcar casi cualquier cosa: entradas de blog, vídeos, audio, imágenes, páginas wiki, perfiles de usuario, marcadores, eventos, etc. Una vez publicado el contenido, el resto de usuarios pueden anotarlo con etiquetas, navegar por temas relacionados a través de ellas, discutir el contenido mediante comentarios o incluso conectarse entre sí directamente o a través del contenido publicado [Bre09].

La adición de etiquetas al contenido publicado forma parte de lo que se denomina “**anotación social**” y se ha convertido en una práctica común y popular en los sitios web sociales. Una etiqueta es un tipo de anotación social, formada por una palabra clave que define un tema o categoría para el contenido asociado, de manera que dicho contenido es más fácil de navegar y descubrir por otros. La adición de anotaciones a los contenidos de las redes sociales (mediante etiquetas de temas, localización geográfica, etc.) es particularmente útil para navegar y localizar elementos interesantes y personas con intereses similares. El éxito de los sitios web sociales se debe al resultado de la creación

¹²Disponible en: <https://forestexplorer.gsic.uva.es/sparql/>

¹³<https://es.wikipedia.org/wiki/Wikipedia:Portada>

de comunidades que se han formado en torno a intereses comunes (como películas, una tecnología en particular, árboles, etc.), en las que los usuarios participan activamente tanto consumiendo información como proporcionando nuevos contenidos y metadatos [Bre09].

A pesar de las ventajas que ofrece la anotación social de elementos de contenido, el etiquetado conduce a diversas limitaciones relacionadas con la recuperación y reutilización de la información publicada [Mat04]. Debido a que los motores de búsqueda dedicados a las etiquetas se basan tan solo en cadenas de texto plano, al realizar una búsqueda con una determinada etiqueta, se obtendrá solo el contenido que ha sido etiquetado con esa palabra clave en particular. Los problemas derivados de esto se describen a continuación:

- **Sinonimia** o ambigüedad de la etiqueta [Wu06]. Como las etiquetas son solo cadenas de texto que carecen de semántica, una máquina no sabrá diferenciar, en muchas ocasiones, palabras que significan cosas diferentes dependiendo del contexto en el que se encuentran. Como es el caso de “Apple”, que podría referirse a la fruta o a la marca de dispositivos inteligentes.
- **Polisemia** o heterogeneidad de la etiqueta. Se pueden utilizar diferentes etiquetas para referirse a una misma cosa, por lo que será necesario realizar varias búsquedas para obtener todo el contenido relacionado con un concepto u objeto en particular [Wu06].
- Falta de organización entre las etiquetas. Las etiquetas carecen de relaciones entre ellas (al ser texto plano), lo que dificulta la búsqueda de información si no se está mirando directamente a la etiqueta adecuada. Esto es claramente un problema, pues los usuarios, dependiendo de su nivel de experiencia, pueden utilizar términos más o menos amplios tanto para etiquetar nuevos conceptos u objetos, como para realizar búsquedas [Bre09].

Estos problemas derivan en anotaciones que solo sirven de utilidad en el sitio web en el que son creadas, difíciles de acceder por otros medios y, consecuentemente, resulta complicado reutilizar la información que éstas aportan. La manera de solventar esta situación propone la modelización de las anotaciones sociales mediante tecnologías de la Web Semántica, dando lugar a lo que se denominan “**anotaciones socio-semánticas**”.

2.3.1. Anotación sociosemántica

La anotación socio-semántica consiste en utilizar las tecnologías semánticas para representar las anotaciones realizadas por los usuarios en la Web [Wu06]. Estos nuevos modelos tratan de solventar los problemas descritos anteriormente y traen consigo tres claros beneficios: (i) permiten capacidades de consulta de contenido más avanzadas, (ii) vincular los datos de las anotaciones a otros datos de la Web semántica y (iii) reutilizarlos a través de otras aplicaciones [Bre09]. Éste último permite a su vez la ampliación de la nube de Datos Abiertos Enlazados (ver Figura 2.3).

Existen vocabularios establecidos que pueden permitir el intercambio de conocimiento colaborativo usando la Web Semántica, como es el caso de FOAF¹⁴ (*Friend Of A Friend*, que significa “Amigo de un Amigo”) pues sirve para describir a las personas, sus actividades y sus relaciones con otras personas y objetos. Este vocabulario se utiliza, por ejemplo, en

¹⁴<http://xmlns.com/foaf/spec/>

la ontología de la aplicación web diseñada en este Trabajo Fin de Grado para declarar a los diferentes usuarios y definir sus propiedades: nombre, nombre de usuario y correo electrónico. De esta manera, cada usuario de la aplicación se identifica como clase `foaf:Person`, que tiene algunas propiedades como `foaf:name` “Jimena Andrade”, `foaf:nick` “jimena22” y `foaf:mbox` “jimena22@gmail.com”. A su vez, las anotaciones de árboles que se crean en la aplicación tienen una propiedad (`dc:creator`) que permiten asociar la información añadida a cada usuario. Existen multitud de clases y propiedades más relacionadas con los usuarios, se pueden consultar todas ellas en su sitio web. Es importante destacar que el uso de ontologías ampliamente aceptadas es fundamental para fomentar la interoperabilidad y reutilización de los datos (y, de esa manera, conseguir conjuntos de datos de “5 estrellas”).

Esta forma de organizar los datos sociales y la información generada por ellos favorece la creación de una “**Web Semántica Social**”, donde todos los diferentes sistemas de colaboración como blogs, wikis y aplicaciones sociales, están conectados entre sí mediante la adición de semántica, permitiendo a los usuarios atravesar cualquier tipo de sistema y reutilizar los datos entre ellos según sea necesario [Bre09]. Es por ello que, el reto de este tipo de aplicaciones sociales y semánticas recae en gran medida en un buen diseño de la interfaz de usuario e interconexión con el *backend* semántico, ya que el principal objetivo de las mismas es que usuarios no expertos en tecnologías semánticas puedan utilizar datos disponibles en formatos de LOD.

2.4. Aplicaciones de datos abiertos en el ámbito forestal

El campo de la ingeniería forestal no se ha quedado atrás en esta nueva forma de crear conocimiento, a través de la colaboración ciudadana y reutilización de datos publicados. Existen algunas aplicaciones relacionadas con este campo, pero no todas ellas hacen uso de Datos Abiertos. En Castilla y León, por ejemplo, existe una aplicación que sí hace uso de ellos: “Incendios CyL¹⁵”, una herramienta gratuita que ofrece información actualizada para la prevención de incendios forestales y de más de 300 infraestructuras recreativas de la comunidad.

A continuación se centra el estudio en dos aplicaciones populares en este ámbito. La primera de ellas ha sido escogida por la importancia que presenta a la hora de consultar datos abiertos forestales y la segunda por ser una aplicación altamente impulsada por la colaboración ciudadana.

2.4.1. *Forest Explorer*

El explorador forestal¹⁶ es una aplicación web creada por investigadores procedentes de diferentes grupos de investigación de la Universidad de Valladolid (GSIC y iuFOR) que permite navegar fácilmente por el contenido del conjunto de Datos Abiertos forestales generados con el proyecto *Cross-Forest*. Su creación surgió debido a la necesidad de una herramienta más amigable que el almacén de triplas, pues la mayor parte de usuarios interesados en usar estos datos no dominan el lenguaje de la Web Semántica.

¹⁵<https://www.incendioscyl.es/>

¹⁶<https://forestexplorer.gsic.uva.es/explorer/>

El explorador forestal está compuesto por un mapa interactivo que permite al usuario navegar por el área de interés, donde se representa la información forestal de la zona con diferentes niveles de abstracción, según el zum escogido [VG20]. La aplicación sigue en continuo desarrollo y forma parte de las anteriores mencionadas iniciativas *open* del gobierno de España. Sin embargo, es una herramienta tan solo de consulta que no está preparada para la anotación de datos forestales.

2.4.2. *PlantNet*

*PlantNet*¹⁷ es una aplicación desarrollada en el marco de un proyecto internacional basado en la ciencia ciudadana que ayuda a identificar plantas de todo el mundo gracias a imágenes almacenadas en su base de datos. La iniciativa tiene como objetivo promover la cultura científica, técnica e industrial de las ciencias de las plantas y cerrar la brecha entre la sociedad y éstas. La aplicación ofrece la posibilidad de identificar a través de un buscador plantas silvestres y ornamentales. Para identificar una planta, *PlantNet* dispone de una base de datos con imágenes cada vez más amplia. Gracias a la colaboración de los diferentes usuarios, en 2019 su base de datos contaba ya con 20000 especies y más de 1 millón de imágenes subidas. Los desarrolladores del proyecto revisan cada contribución para que la información sea fiable y lo más segura posible.

Este proyecto toma cierta relevancia debido a que se basa en un modelo colaborativo de la ciudadanía (usa anotación social), difundiendo el conocimiento libre y poniendo una herramienta de bajo coste al alcance de cualquiera. Además, los datos generados están accesibles para todo el público general, promoviendo el movimiento de Datos Abiertos. Sin embargo, *PlantNet* no usa tecnologías semánticas para estructurar estos datos, por lo que no resulta fácil su reutilización al no formar parte de un conjunto de datos abiertos enlazados.

2.5. Discusión y conclusiones

La participación activa de las administraciones en la publicación de datos en la Web está generando un sinnúmero de conjuntos de datos que pueden utilizarse a favor de sectores económicos, sociales o académicos. El problema de estos conjuntos de datos es que carecen de estructuras definidas, dificultando su identificación por las máquinas y reutilización. Esto provoca que, en la mayoría de los casos, los datos publicados pasen a crear silos y vertederos en la Web. El movimiento de datos abiertos enlazados (LOD) pretende solventar estos problemas. Un caso de éxito de esta aproximación es el *Cross-Forest*, un proyecto que centra sus esfuerzos en transformar los datos inventarios forestales en Datos Abiertos Enlazados.

Por otro lado, se ha analizado la popularidad que las aplicaciones web en el sector forestal están teniendo. Estas aplicaciones están promovidas por la participación activa de los ciudadanos, quienes anotan elementos relacionados con los árboles y las plantas. El problema clave es que estas anotaciones y la información que se extrae de ellas es difícil de clasificar, pues las anotaciones se hacen en texto plano y, por consiguiente, de reutilizar en otras aplicaciones. Por ello, se puede recurrir a las tecnologías de la Web Semántica para definir modelos de los datos que los ciudadanos anotan. Estos datos se beneficiarían de los

¹⁷<https://plantnet.org/en/>

aspectos positivos de la Web Semántica, además de hacer crecer la nube de datos abiertos enlazados.

Tras este análisis, cabe proponer una solución para el campo de la ingeniería forestal en la que explotar y complementar datos abiertos mediante la ciencia ciudadana. En otras palabras, lo que se plantea en este TFG es crear una aplicación web social que haga uso de ontologías (existentes o nuevas, pero relacionadas entre sí) de tal manera que se consiga (i) aprovechar la ciencia ciudadana para facilitar la generación de datos forestales, (ii) favorecer la reutilización de los datos creados gracias a la Web Semántica y (iii) fomentar el uso de datos abiertos de ámbito forestal con propósito formativo. Esta aplicación, además de cumplir con los objetivos que persigue la utilización de la Web Semántica, debe cumplir otros tales como el curado de datos (comprobación de que los datos subidos son correctos). Esto último se pretende conseguir gracias a la colaboración ciudadana, los usuarios serán quienes validen los diferentes datos de la aplicación web.

Es dentro de tal propuesta donde surge este Trabajo Fin de Grado, que se centra únicamente en el diseño de la interfaz de usuario (*frontend*). La ontología utilizada en la aplicación se ha creado específicamente para este fin, aprovechando ciertos aspectos técnicos y reutilizando datos provenientes del proyecto *Cross-Forest*, en concreto todos los árboles registrados en la provincia de Valladolid y toda la información relacionada con las especies. Además, los datos generados en la aplicación harán uso de un *backend* diseñado por Cristina Mayo y que se puede consultar en [May20]. En las secciones posteriores se analizarán más en profundidad los requisitos y aspectos técnicos de la aplicación, siempre teniendo en cuenta que el objetivo final perseguido es el que se explica en esta sección.

Capítulo 3

Análisis

3.1. Introducción

En la Figura 3.1 se presenta una visión de la aplicación denominada *Timber* en su totalidad, donde se plantea una arquitectura multicapa compuesta por el *frontend* (rectángulo (a)) y el *backend* (rectángulo (b)). El ámbito de este TFG abarca el rectángulo (a), por lo que el presente capítulo se centra en el análisis del *frontend* de la aplicación web. Al hablar de “el sistema”, se hace referencia a dicha parte y no a la aplicación completa. El *backend* de [May20] pasa a ser un actor más dentro del sistema y, como se ha comentado en capítulos anteriores, los datos manejados son estructurados según la Ontología del Apéndice A.

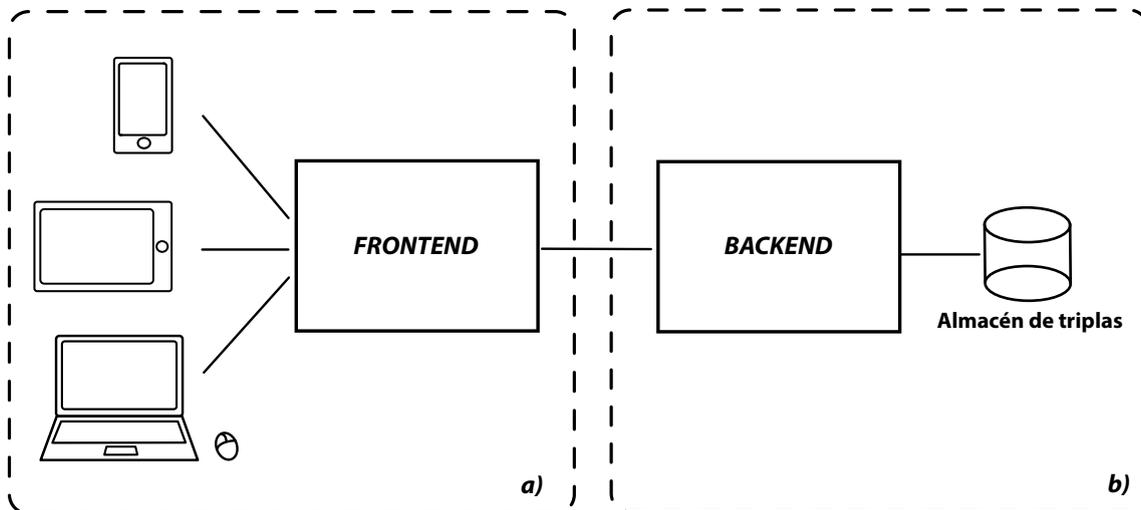


Figura 3.1: Visión de la aplicación completa

Este capítulo es el resultado de las fases de arranque y elaboración de la metodología del Proceso Unificado indicada en la sección 1.3 de este documento. El análisis consta de tres partes fundamentales. El apartado 3.2 está dedicado al documento de **visión del sistema**, que constituye una descripción informal del sistema software, en este caso la aplicación web, donde se describen los principales aspectos de la misma. La recopilación de **requisitos** del sistema se realiza en la sección 3.3. Esta parte del proceso se considera la más importante dentro del análisis, puesto que si los requisitos están mal definidos, la aplicación no cumplirá con las necesidades de los usuarios. Los requisitos recogen las

condiciones que debe cumplir o poseer el sistema o uno de sus componentes para satisfacer una determinada especificación. Además, sirven de mucha utilidad a lo largo del proceso de desarrollo del software: al principio se recopilan para determinar lo que se va a construir, a lo largo del desarrollo sirven para guiar el proceso y al final del proyecto, se utilizan para verificar que la aplicación final hace realmente lo que se suponía que debía hacer. Finalmente, en la sección 3.4, se describen los **casos de uso** del sistema. Un caso de uso es una descripción de una serie de interacciones entre actores, que permite capturar las funcionalidades del sistema. La descripción de los mismos está acompañada por un Diagrama de Casos de Uso, que representa la manera en la que los actores utilizan la aplicación [Ste15].

El capítulo finaliza con un apartado dedicado al proceso que se siguió para realizar el **prototipo de la interfaz** de usuario en papel y una última sección que trata el problema relacionado con la **validación de los datos** mediante la colaboración de la comunidad de usuarios. Además, en el Apéndice B se especifica un **glosario**, formado por una lista de los términos más relevantes y sus definiciones, cuyo objetivo es que el lector logre un buen conocimiento del dominio en el que se enmarca la aplicación *Timber*.

3.2. Visión del sistema

La herramienta *Timber* pretende ser una aplicación web de carácter social, pensada para uso en dispositivos móviles, que facilite el intercambio de información relativa a los árboles entre los usuarios de la plataforma y, a su vez, esos datos formen parte de la nube de datos abiertos enlazados. Los usuarios de la aplicación pueden anotar árboles, geolocalizarlos y añadir características de los mismos, además de compartirlas con el resto de usuarios de una manera intuitiva y cómoda. La aplicación cuenta con datos procedentes del conjunto de datos abiertos del Inventario Forestal Nacional (IFN) y el Mapa Forestal de España (MFE). De manera transparente al usuario, la aplicación estructura la información publicada con vocabularios de la Web Semántica, generando un conjunto de datos abiertos enlazados.

Timber está pensada para cualquier ámbito de uso, centrandolo en el ámbito educativo y divulgativo. Se espera que la aplicación pueda ser utilizada en las aulas, como medio tecnológico para aprender y difundir conocimientos sobre los árboles de una manera novedosa, aunque también se pretende que tenga cabida en usuarios aficionados, amantes de los árboles o del sector forestal en general. No se descarta su uso “profesional” para tareas de actualización y mantenimiento de catálogos forestales, entre otras.

Dentro de la aplicación, existen dos tipos de usuarios:

- **No registrado:** aquel usuario que no desea registrarse. Este tipo de usuario puede acceder a la aplicación y navegar por el mapa para ver la información almacenada, sin necesidad de contribuir en ella.
- **Registrado:** usuario que entra en la aplicación mediante un nombre de usuario y una contraseña. Previamente ha tenido que darse de alta en el sistema. Este usuario puede crear árboles nuevos, completar información de árboles ya existentes mediante anotaciones, ver las actividades de otros usuarios, debatir sobre la información publicada, etc. En definitiva, contribuir al crecimiento de la aplicación.

Dentro de los **usuarios registrados**, se establece una subclasificación en 3 niveles (que servirá principalmente para establecer las bases en la validación de los datos introducidos). Esta subclasificación se basa en el número de puntos que posee un usuario. Los puntos se consiguen mediante la participación activa en la aplicación, es decir, consiguiendo **logros**. Los puntos obtenidos con los logros pretenden “medir” relativamente el conocimiento de los usuarios con respecto al campo de los árboles. De tal manera que, perteneciendo al nivel más alto, las aportaciones serán tomadas más en cuenta por parte de la aplicación. Los 3 niveles se especifican a continuación:

1. **Principiante**: nivel más bajo
2. **Aficionado**: nivel intermedio
3. **Experto**: nivel más alto

Inicialmente, todos los usuarios registrados forman parte del nivel **Principiante**. En la sección 3.6 se detalla cómo se pueden obtener puntos y los requisitos necesarios para poder subir de nivel.

Tan solo es necesario un dispositivo con acceso a Internet y un navegador moderno para poder acceder a la aplicación. La interfaz de usuario será *responsive*¹ o adaptativa, de manera que se ajuste automáticamente a las características de visualización (resolución, tamaño de pantalla, vistas) del dispositivo empleado. Su prototipo se especifica en profundidad en el apartado 3.5 de este capítulo.

3.3. Definición de requisitos

De manera general, los requisitos se pueden clasificar como Funcionales (FRQ, *functional requirement*) y No Funcionales (NFRQ, *non-functional requirement*) [Ste15]. Un requisito funcional es una característica requerida del sistema que expresa una funcionalidad del mismo. A su vez, un requisito no funcional representa una característica requerida del sistema que señala una restricción del mismo. Para abarcar todos los requerimientos del sistema, los diferentes requisitos redactados han sido diseñados con la categorización **FURPS+** [Ste15], acrónimo de las siguientes categorías (y que recoge tanto requisitos funcionales como no funcionales):

- **Funcionalidad** (*Functionality*): expresan qué debe hacer la aplicación.
- **Usabilidad** (*Usability*): cómo debe ser la aplicación. Están orientados al usuario, como la facilidad de uso, la apariencia general, el modo de navegación, etc.
- **Fiabilidad** (*Reliability*): cómo de fiable debería ser el sistema.
- **Rendimiento** (*Performance*): expresa cuál debería ser la eficiencia de la aplicación.
- **Soporte** (*Supportability*): cómo de fácil es dar soporte a la aplicación.
- **El signo +**: hace referencia a requisitos adicionales que suelen ser restricciones. Incluyen restricciones de diseño, requisitos de implementación, requisitos de interfaz y restricciones físicas.

¹https://www.w3schools.com/html/html_responsive.asp

A continuación se especifican los diferentes requisitos recogidos durante las primeras semanas de reuniones, en las que participaron tanto los tutores de este TFG, como otros profesores del departamento y Cristobal Ordóñez, un especialista en la gestión de datos forestales.

3.3.1. Requisitos Funcionales

FRQ001. El sistema deberá facilitar a los usuarios darse de alta en la aplicación, de manera que pasen a ser usuarios registrados. La información necesaria para el registro será: el nombre completo, el nombre de usuario (o *nickname*), la contraseña y una dirección de correo electrónico. El sistema deberá mandar los datos al *backend* para su almacenamiento.

FRQ002. El sistema deberá impedir la creación de más de una cuenta asociada a un mismo nombre de usuario o correo electrónico. Indicando al usuario que ya existe otra cuenta con dicha información.

FRQ003. El sistema deberá facilitar a los usuarios tanto iniciar como cerrar la sesión. Para el inicio de sesión será necesario introducir el *nickname* y la contraseña. El sistema mandará los datos introducidos al *backend* para comprobar si son correctos.

FRQ004. El sistema permitirá a los usuarios darse de baja de la aplicación. En tal caso, las anotaciones realizadas permanecerían en la aplicación.

FRQ005. El sistema deberá obtener del *backend* la información personal de los usuarios, visualizarla y, si fuera necesario, permitir su modificación. Esta información incluye el nombre completo, el *nickname*, el correo electrónico, el nivel actual al que pertenece y el número de puntos conseguidos hasta el momento.

FRQ006. El sistema deberá obtener del *backend* los puntos y el nivel actual de cada usuario e informarle de su nivel actual y los puntos necesarios para subir de nivel.

FRQ007. El sistema deberá transformar todos los datos de entrada del usuario (información personal y relativa a los árboles) en llamadas a la API del *backend* para que puedan ser almacenados correctamente según la ontología diseñada.

FRQ008. El sistema deberá facilitar a los usuarios registrados dar de alta un nuevo árbol en la aplicación, siendo necesario únicamente especificar la localización del mismo. Adicionalmente se podrá incluir la especie y añadir una imagen del mismo. El sistema mandará dichos datos al *backend* para su almacenamiento.

FRQ009. El sistema permitirá a los usuarios registrados crear anotaciones nuevas en un árbol existente de la aplicación. El sistema deberá enviar al *backend* los nuevos datos para su almacenamiento.

FRQ010. El sistema deberá obtener del *backend* la información de todos los árboles y anotaciones creadas por un usuario registrado y facilitar su visualización.

FRQ011. El sistema facilitará a los usuarios registrados la posibilidad de expresar su acuerdo o desacuerdo con las anotaciones de los árboles realizadas por otros usuarios registrados y mandar dicha información al *backend* para su almacenamiento.

FRQ012. El sistema deberá comunicarse con el *backend* para facilitar a los usuarios registrados modificar anotaciones creadas por ellos mismos.

FRQ013. El sistema permitirá a los usuarios registrados eliminar anotaciones creadas por ellos mismos y comunicárselo al *backend* para poder eliminarlas de la aplicación.

FRQ014. El sistema deberá permitir visualizar todos los árboles almacenados en el *backend* a cualquier usuario que entre en la aplicación.

FRQ015. El sistema deberá permitir a los usuarios la navegación por un mapa que muestre los árboles de la aplicación. El sistema deberá obtener la información de los árboles del *backend*.

FRQ016. El sistema deberá obtener del *backend* los detalles de un árbol seleccionado y mostrarlos en la interfaz. Estos incluirán: especie y localización primarias, fecha de creación y anotaciones asociadas (de imagen, especie y localización) de otros usuarios.

FRQ017. El sistema deberá obtener del *backend* las anotaciones con más valoraciones positivas y mostrarlas indicando que son las anotaciones más populares por parte de los usuarios, también llamadas anotaciones primarias.

FRQ018. El sistema deberá permitir realizar búsquedas de árboles según una determinada especie, un determinado creador o una localización específica.

FRQ019. El sistema deberá obtener del *backend* los árboles filtrados según una determinada especie, un determinado creador o una localización específica.

FRQ020. El sistema deberá obtener del *backend* la información relacionada con las actividades de otros usuarios registrados del sistema y facilitar su visualización.

FRQ021. El sistema facilitará la interacción entre los usuarios registrados de la aplicación.

3.3.2. Requisitos no Funcionales

NFRQ001. El sistema deberá ser de uso fácil, de manera que cualquier usuario sin conocimientos técnicos de bases de datos con ontologías ni forestales pueda dominarlo al poco tiempo de acceder a la aplicación. La interfaz debe mostrar claramente las diferentes funcionalidades que se ofrecen.

NFRQ002. El sistema deberá visualizarse y funcionar correctamente en la última versión de los navegadores Mozilla Firefox, Google Chrome y Safari.

NFRQ003. La interfaz de usuario deberá adaptarse adecuadamente a cualquier dispositivo utilizado (ordenadores de sobremesa, móviles y tabletas).

NFRQ004. El sistema deberá presentar inicialmente la interfaz al usuario en el idioma español, aunque además dicha interfaz ha de estar disponible en inglés.

NFRQ005. El sistema deberá solucionar localmente el hecho de que se produzca algún fallo al usar un servicio externo (como el mapa o la API REST) e informar adecuadamente al usuario.

NFRQ006. El sistema permitirá una navegación fluida por el mapa.

NFRQ007. El sistema no deberá tardar más de 15 segundos en mostrar los resultados de una búsqueda de árboles. Si se supera este plazo, el sistema indicará que hay problemas de conexión con el servidor.

NFRQ008. El sistema deberá contar con un sencillo manual de uso que explique de manera breve las principales acciones disponibles en la aplicación.

NFRQ009. El sistema deberá garantizar la legibilidad de la interfaz para los diferentes dispositivos desde los cuales se puede acceder al mismo. Para ello se impondrá un tamaño de fuente visible cómodamente desde una distancia mínima de 65 centímetros.

NFRQ010. El sistema solo estará disponible para aquellos dispositivos con conexión adecuada a Internet y con un navegador moderno.

3.4. Casos de Uso

Se pueden diferenciar varios tipos de funcionalidades que ofrece la aplicación. Por simplicidad en la descripción de los mismos, se han dividido los Diagramas de Casos de Uso en tres: el primero de ellos corresponde a los casos de uso relacionados con la gestión de los usuarios, el segundo con la gestión de los árboles y anotaciones y, el tercero, con la navegación y la búsqueda de contenidos.

3.4.1. Gestión de usuarios

La Figura 3.2 muestra el diagrama de casos de uso relacionados con la gestión de los usuarios dentro del sistema y que se corresponden con los requisitos funcionales FRQ001 al FRQ007 (ver sección 3.3.1). La descripción de los casos de uso se detalla a continuación.

Caso de uso: *DarseDeAlta*

Descripción breve. Un usuario se registra por primera vez en la aplicación.

Precondición. Ninguna.

Descripción paso a paso.

1. El usuario no registrado selecciona la opción “Unirse” del menú de navegación.
2. El sistema solicita el nombre completo, el *nickname*, el correo electrónico y la contraseña.
3. El usuario introduce los datos solicitados.
4. El sistema manda los datos al *backend*.
5. El *backend* comprueba que no existe ya ningún usuario con ese correo electrónico ni nombre de usuario y, en caso afirmativo, crea la cuenta de usuario. En caso contrario, solicita al sistema que el usuario introduzca unos datos nuevos.
6. Una vez creada la cuenta, el sistema traslada al usuario a la página principal de usuarios registrados.

Postcondición. El usuario ha sido creado en la aplicación.

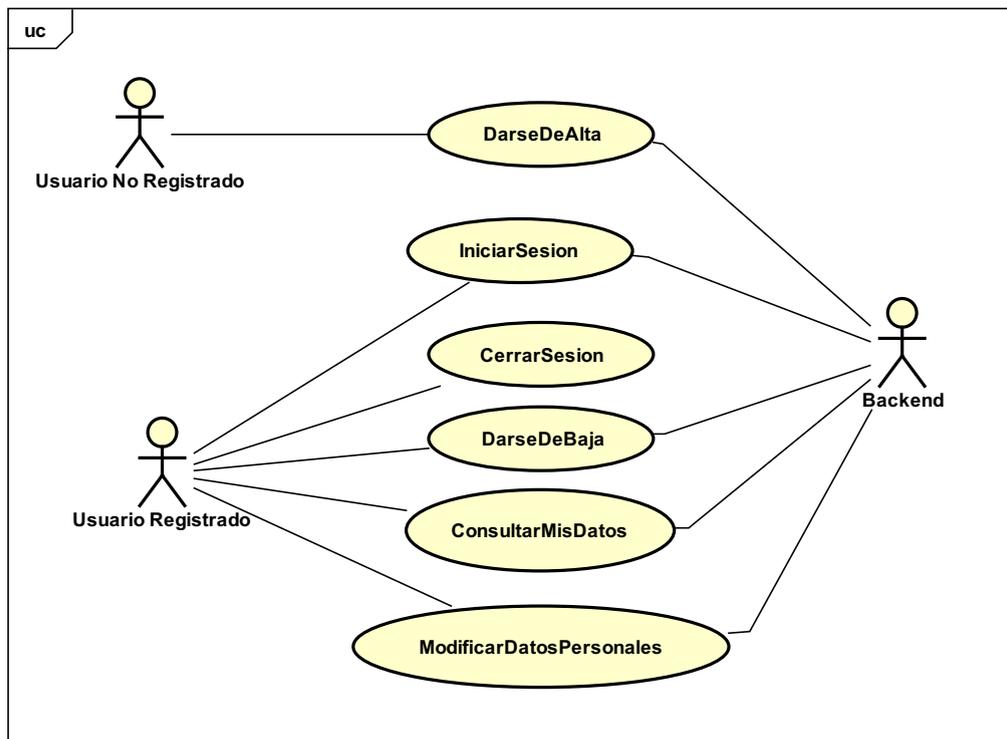


Figura 3.2: Diagrama de Casos de Uso: gestión de usuarios

Caso de uso: *IniciarSesion*

Descripción breve. Un usuario valida su identidad en la aplicación.

Precondición. El usuario está registrado en el sistema.

Descripción paso a paso.

1. El usuario selecciona la opción “Iniciar Sesión” del menú de navegación.
2. El sistema solicita el *nickname* y la contraseña.
3. El usuario introduce los datos solicitados.
4. El sistema envía los datos introducidos al *backend* para comprobar que son correctos.
5. El *backend* manda la confirmación o negación del acceso con los datos del usuario introducidos.
6. El sistema, en caso afirmativo, traslada al usuario a la página principal de usuarios registrados. En caso contrario, solicita al usuario volver a introducir los datos.

Postcondición. El usuario ha iniciado sesión.

Caso de uso: *CerrarSesion*

Descripción breve. Un usuario quiere cerrar sesión en la aplicación.

Precondición. El usuario está registrado en el sistema y ha iniciado sesión.

Descripción paso a paso.

1. El usuario selecciona la opción “Cerrar Sesión” del menú de navegación.
2. El sistema cierra la sesión actual y traslada al usuario a la página de inicio del sistema.

Postcondición. El usuario ha cerrado su sesión.

Caso de uso: *DarseDeBaja*

Descripción breve. Un usuario desea darse de baja de la aplicación.

Precondición. El usuario está registrado en el sistema y ha iniciado sesión.

Descripción paso a paso.

1. El usuario selecciona la opción “Darse de baja del sistema” dentro del apartado de Ajustes.
2. El sistema pregunta al usuario si está seguro que quiere darse de baja, indicándole que toda su información personal se borrará de la aplicación.
3. El usuario confirma la acción.
4. El sistema solicita al *backend* borrar al usuario del registro de usuarios.
5. El *backend* elimina al usuario y su información personal.

Postcondición. El usuario y toda su información personal han sido eliminadas de la aplicación.

Caso de uso: *ConsultarMisDatos*

Descripción breve. Un usuario desea consultar sus datos personales, así como su nivel actual y puntos conseguidos.

Precondición. El usuario está registrado en el sistema y ha iniciado sesión.

Descripción paso a paso.

1. El usuario accede al apartado de “Ajustes” de la aplicación.
2. El sistema solicita al *backend* los datos personales del usuario.
3. El *backend* proporciona los datos solicitados.
4. El sistema muestra los datos personales del usuario (nombre completo, nombre de usuario y correo electrónico), la fecha en la que se registró en la aplicación, los puntos conseguidos con los logros y el nivel actual en el que se encuentra.

Postcondición. Ninguna.

Caso de uso: *ModificarDatosPersonales*

Descripción breve. Un usuario desea modificar sus datos personales guardados en la aplicación.

Precondición. El usuario está registrado en el sistema y ha iniciado sesión.

Descripción paso a paso.

1. El usuario accede al apartado de “Ajustes”.
2. El sistema solicita al *backend* los datos personales del usuario.
3. El *backend* proporciona los datos solicitados.
4. El sistema muestra los datos devueltos dentro del apartado “Datos Personales”.
5. El usuario modifica los datos que desea y pulsa “Modificar datos”.
6. El sistema recoge las modificaciones realizadas y las manda al *backend*.
7. El *backend* registra los datos modificados.
8. El sistema confirma la acción y muestra por pantalla los datos actualizados.

Postcondición. El usuario ha podido actualizar su información personal registrada en la aplicación.

3.4.2. Gestión de árboles y anotaciones

La Figura 3.3 muestra el diagrama de casos de uso relacionados con los árboles y las anotaciones del sistema. Se corresponden con los requisitos funcionales FRQ007 al FRQ0014 (ver sección 3.3.1) La descripción de los casos de uso se detalla a continuación.

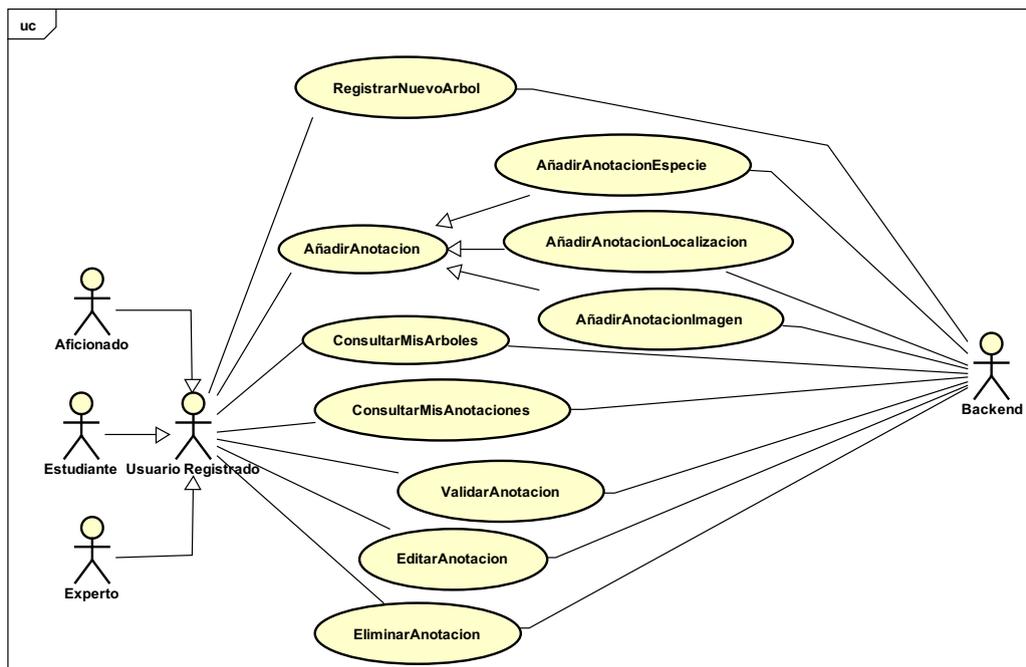


Figura 3.3: Diagrama de Casos de Uso: gestión de árboles y anotaciones

Caso de uso: RegistrarNuevoArbol

Descripción breve. Un usuario registrado desea añadir un nuevo árbol en la aplicación.

Precondición. El usuario ha iniciado sesión como Usuario Registrado.

Descripción paso a paso.

1. El usuario selecciona la opción “Crear nuevo árbol” dentro de la barra de navegación.
2. El sistema solicita la localización del árbol como dato obligatorio para su creación. Adicionalmente se puede incluir la especie y una imagen.
3. El usuario introduce la localización (y el resto de campos opcionalmente).
4. El sistema comprueba que se ha introducido la localización correctamente y, en caso afirmativo, procesa los datos y muestra una previsualización del árbol a crear.
5. El usuario confirma la acción.
6. El sistema adecúa los datos y los envía al *backend* para su almacenamiento.
7. El *backend* confirma que la acción se ha completado con éxito.

Postcondición. Un nuevo árbol ha sido registrado en el sistema.

Caso de uso: AñadirAnotacion

Descripción breve. El usuario está viendo la información de un árbol concreto y desea añadir una nueva anotación.

Precondición. El usuario ha iniciado sesión como Usuario Registrado.

Descripción paso a paso.

1. El usuario está visualizando la información completa de un árbol y selecciona la opción “Añadir anotación”.
2. El sistema solicita al usuario que elija el tipo de anotación que desea realizar: anotación de especie, de localización o de imagen.
3. El usuario selecciona el tipo de anotación que desea realizar.
 - a) El usuario selecciona anotación de especie.
 - b) El usuario selecciona anotación de localización.
 - c) El usuario selecciona anotación de imagen.
4. El sistema solicita los datos necesarios.
 - a) Para la anotación de especie, muestra una lista con todas las especies disponibles en el sistema.
 - b) Para la anotación de localización, solicita las coordenadas geográficas.
 - c) Para la anotación de imagen, solicita la imagen y, adicionalmente, una descripción de la misma, un título y la parte del árbol que muestra.
5. El usuario introduce los campos solicitados y pulsa a “Confirmar anotación”.

6. El sistema adecúa los datos introducidos y los envía al *backend* para su almacenamiento.
7. El *backend* confirma que la acción se ha completado con éxito.

Postcondición. Se ha añadido una nueva anotación a un árbol del sistema.

Caso de uso: *ConsultarMisArboles*

Descripción breve. Un usuario registrado desea ver todos los árboles que ha creado desde su inicio en la aplicación.

Precondición. El usuario ha iniciado sesión como Usuario Registrado.

Descripción paso a paso.

1. El usuario selecciona la opción “Mis árboles” dentro de la barra de navegación.
2. El sistema solicita los árboles creados por el usuario al *backend*.
3. El *backend* proporciona los datos.
4. El sistema muestra un historial en forma de lista con todos los árboles que el usuario ha creado desde que se registró en la aplicación.

Postcondición. Ninguna.

Caso de uso: *ConsultarMisAnotaciones*

Descripción breve. Un usuario registrado desea ver todas las anotaciones que ha creado desde su inicio en la aplicación.

Precondición. El usuario ha iniciado sesión como Usuario Registrado.

Descripción paso a paso.

1. El usuario selecciona la opción “Mis anotaciones” dentro de la barra de navegación.
2. El sistema solicita las anotaciones creadas por el usuario al *backend*.
3. El *backend* proporciona los datos.
4. El sistema muestra un historial en forma de lista con todas las anotaciones creadas desde que se registró en la aplicación.

Postcondición. Ninguna.

Caso de uso: *ValidarAnotacion*

Descripción breve. Un usuario está visualizando la información completa de un árbol y desea indicar que está de acuerdo (o en desacuerdo) con la información publicada en una de las anotaciones creadas por otros usuarios.

Precondición. El usuario ha iniciado sesión como Usuario Registrado.

Descripción paso a paso.

1. Un usuario selecciona el árbol que desea consultar.

2. El sistema solicita los datos relacionados con ese árbol al *backend*.
3. El *backend* proporciona los datos.
4. El sistema muestra todas las anotaciones que tiene el árbol. Cada anotación dispone de un apartado que permite al usuario indicar si está de acuerdo o en desacuerdo con dicha anotación.
5. El usuario selecciona una de las opciones de una anotación del árbol.
6. El sistema recoge la selección del usuario y la envía al *backend*.
7. El *backend* registra el voto y actualiza el número de votos de la anotación seleccionada. A continuación, comprueba si esa anotación es la primaria. En caso de que no lo sea, el sistema comprueba si el número de votos supera a los de la anotación primaria, para actualizarla en caso afirmativo.

Postcondición. El usuario ha validado una anotación de un árbol.

Caso de uso: *EditarAnotacion*

Descripción breve. Un usuario desea modificar una anotación creada por él mismo.

Precondición. El usuario ha iniciado sesión como Usuario Registrado.

Descripción paso a paso.

1. El usuario se encuentra consultando su lista de anotaciones.
2. El sistema muestra, al lado de cada anotación, la opción de editarla.
3. El usuario selecciona la opción de “Editar Anotación”.
4. El sistema facilita un campo para introducir la nueva información.
5. El usuario introduce los nuevos datos y confirma la edición.
6. El sistema recoge los datos modificados y los envía al *backend* para su almacenamiento.
7. El *backend* registra las modificaciones y envía la confirmación de la acción.
8. El sistema muestra de nuevo la lista de anotaciones actualizada.

Postcondición. El usuario ha modificado una anotación realizada por él mismo.

Caso de uso: *EliminarAnotacion*

Descripción breve. Un usuario desea eliminar una anotación creada por él mismo.

Precondición. El usuario ha iniciado sesión como Usuario Registrado.

Descripción paso a paso.

1. El usuario se encuentra consultando su lista de anotaciones.
2. El sistema facilita, al lado de cada anotación, la opción de eliminarla.
3. El usuario selecciona la opción de “Eliminar Anotación”.
4. El sistema indica al usuario si desea eliminarla definitivamente de la aplicación.
5. El usuario confirma la acción.
6. El sistema recoge la información y la envía al *backend*.

7. El *backend* elimina la anotación y envía la confirmación de la acción.
8. El sistema muestra la lista de anotaciones actualizada.

Postcondición. El usuario ha eliminado una anotación realizada por él mismo.

3.4.3. Navegación y búsqueda

La Figura 3.4 muestra el diagrama de casos de uso relacionados con la navegación de los árboles del sistema y la búsqueda especializada de los mismos. Se corresponden con los requisitos funcionales FRQ015 al FRQ019 (ver sección 3.3.1). La descripción de los casos de uso se detalla a continuación.

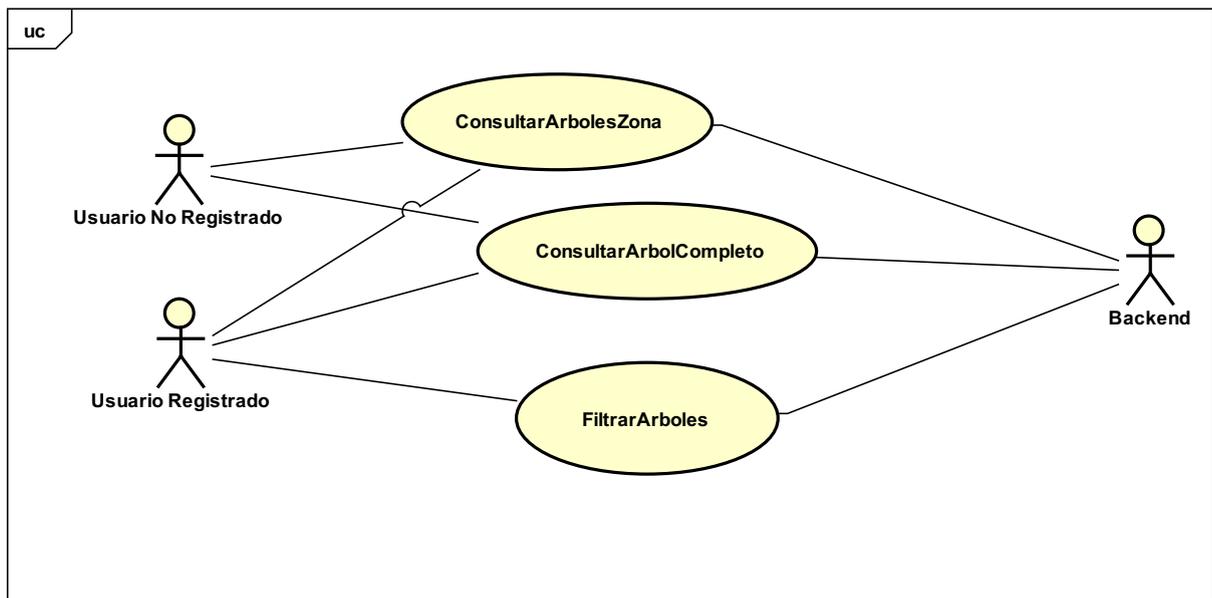


Figura 3.4: Diagrama de Casos de Uso: navegación y búsqueda

Caso de uso: *ConsultarArbolesZona*

Descripción breve. Un usuario de la aplicación desea ver los árboles que hay registrados en una determinada zona.

Precondición. Ninguna.

Descripción paso a paso.

1. El sistema muestra un mapa ubicado en Valladolid con los árboles que hay registrados en el *backend* en la zona que se muestra.
2. El usuario se sitúa en la zona donde desea consultar los árboles.
3. El sistema solicita al *backend* los árboles que existen en la zona del mapa donde se encuentra situado el usuario.
4. El *backend* proporciona los datos solicitados.
5. El sistema pinta en el mapa los árboles registrados en la aplicación y también da la opción de verlos en “Modo Lista”.

6. El usuario selecciona uno de los árboles mostrados.
7. El sistema muestra la especie y el creador del árbol seleccionado.

Postcondición. El usuario ha consultado los árboles registrados en la aplicación en una zona determinada.

Caso de uso: *ConsultarArbolCompleto*

Descripción breve. Un usuario está consultando la lista de árboles del sistema y desea ver toda la información de uno de ellos en concreto.

Precondición. Ninguna.

Descripción paso a paso.

1. El usuario selecciona un árbol concreto dentro de la lista de árboles.
2. El sistema solicita al *backend* la información de dicho árbol.
3. El *backend* proporciona los datos solicitados.
4. El sistema muestra toda la información almacenada del árbol en cuestión, que incluye: su creador, la especie y localización primarias, la fecha de creación y todas sus anotaciones asociadas.

Postcondición. El usuario ha consultado toda la información de un árbol registrado en la aplicación.

Caso de uso: *FiltrarArboles*

Descripción breve. Un usuario de la aplicación desea consultar árboles que cumplan algún requisito determinado.

Precondición. El usuario ha iniciado sesión como Usuario Registrado.

Descripción paso a paso.

1. El usuario selecciona la opción “Búsqueda” dentro de la barra de navegación.
2. El sistema facilita un formulario donde el usuario puede indicar una especie y/o un creador.
3. El usuario introduce una especie, un creador o ambas y selecciona la opción de “Buscar”.
4. El sistema solicita al *backend* la información deseada.
5. El *backend*, en caso de que existan árboles que cumplen las características seleccionadas, proporciona los datos solicitados.
6. El sistema muestra en forma de lista los árboles filtrados. En caso de que no haya ninguno registrado, indica al usuario que no existen árboles registrados con esas características.

Postcondición. El usuario ha consultado los árboles registrados en la aplicación que cumplan con una determinada especie y/o creador.

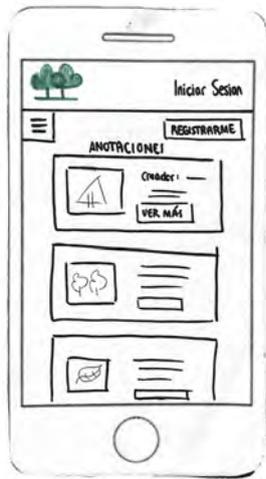
3.5. Prototipado de interfaz de usuario

La creación de prototipos en papel (*paper prototyping*) es un método ampliamente utilizado para diseñar, probar y refinar las interfaces de usuario de una manera rápida y eficiente [Sny03]. Durante la etapa de recopilación de requisitos, se optó por probar esta técnica, ya que es una manera fácil de intercambiar ideas, no solo entre los participantes del proyecto software, sino también con expertos del dominio en el que se enmarca la aplicación. El prototipo en papel se presentó a los tutores del presente TFG junto con otros profesores del mismo departamento y Cristina Mayo. Se contó, además, con la participación de Cristobal Ordóñez que, como se comentó anteriormente, es un experto del dominio forestal y cuya participación resultó muy positiva para el desarrollo de este proceso. Para realizar la presentación, se creó un vídeo, con el fin de comunicar visualmente las ideas, donde se muestra cómo se van desarrollando las diferentes funcionalidades de la interfaz. Dicho vídeo se puede ver en el siguiente enlace:

<https://youtu.be/sWs0EhbtPI4>

Es importante resaltar que el prototipo que se propuso no se centra en el diseño exacto de la interfaz, sino en qué papel juega la misma en la realización de los diferentes casos de uso de la aplicación. Gracias a su puesta en práctica, los participantes en la reunión en la que se presentó la propuesta pudieron aportar sus opiniones y realimentación sobre la idea presentada. Además, el prototipo consiguió unificar ideas entre todos los participantes, aclarar requisitos que no estaban aún claros y descubrir qué partes de la interfaz se adecuaban a las especificaciones y qué carencias presentaba. Por ejemplo, en una primera aproximación del prototipo nos dimos cuenta de que sería útil un mapa para mostrar la localización de los árboles del sistema (aparte de la opción del “Modo Lista”). El diseño inicial de este primer prototipo se muestra en la Figura 3.5. Los casos de uso que representan cada fotografía se especifican a continuación:

- a. Página principal: pertenece al “Modo Lista” del caso de uso *ConsultarArbolesZona*.
- b. Detalles de un árbol: caso de uso *ConsultarArbolCompleto*.
- c. Inicio de sesión: formulario del caso de uso *IniciarSesion*.
- d. Principal usuarios registrados: vista de la sección a la que se accede tras darse de alta en el sistema (*DarseDeAlta*) o iniciar sesión (*IniciarSesion*).
- e. Registrar árbol: vista del formulario para la realización del caso de uso *RegistrarNuevoArbol*.
- f. Mis anotaciones: vista correspondiente al caso de uso *ConsultarMisAnotaciones*.
- g. Búsqueda: vista de la idea inicial del formulario para el caso de uso *FiltrarArboles*.
- h. Anotaciones de un árbol: vista de las anotaciones correspondiente a parte del caso de uso *ConsultarArbolCompleto*. También se refleja la idea de validar una anotación, propia del caso de uso *ValidarAnotacion*.



(a) Página principal



(b) Detalles de un árbol



(c) Inicio de sesión



(d) Principal usuarios registrados



(e) Registrar árbol



(f) Mis anotaciones



(g) Búsqueda



(h) Anotaciones de un árbol

Figura 3.5: Prototipo inicial de la interfaz de usuario

3.6. Propuesta de curado de datos

Al tratarse de una aplicación social, los datos que se introducen pueden no ser válidos, pues los usuarios no tienen por qué ser expertos en el tema. *Timber* no cuenta con ayuda externa para validar la información publicada, sino que, gracias a la colaboración de la comunidad de usuarios, se pretende que los contenidos puedan ser actualizados automáticamente en función de la popularidad de los mismos.

Existen multitud de aplicaciones que se basan en la **colaboración ciudadana** para la verificación de contenidos. Por ejemplo, la aplicación colaborativa *Waze* (comentada en el Capítulo 1 de este documento), divide a los usuarios según niveles. Cuando los usuarios realizan ediciones en el mapa de carreteras, se les otorga diferentes puntos. Una vez alcanzados un número de puntos determinado, los usuarios pueden optar a subir de posición. Esta aplicación sí que está supervisada por administradores que controlan los contenidos publicados y qué usuarios pueden subir de nivel. Otro ejemplo colaborativo son las *Comunidades de Google*, espacios donde los usuarios pueden hacer preguntas o añadir comentarios sobre los productos y servicios de esta empresa, dar consejos a la comunidad acerca de su uso y obtener ayuda de otros usuarios y entusiastas de Google. La aplicación web no supervisa de forma proactiva el contenido de la comunidad, así que es posible que el contenido no sea correcto o que no esté actualizado. Como método sustitutivo, existen las “Respuestas recomendadas”, que son las respuestas que los usuarios consideran correctas o útiles. Estas respuestas pueden estar marcadas por Expertos (plata, oro y platino) o representantes de Google, cuyos comentarios están identificados por una insignia para facilitar al lector saber quién ha validado la información expuesta.

Tras el análisis de los métodos utilizados por estas populares aplicaciones, se ha diseñado uno propio para la aplicación de *Timber*. Para ello, los usuarios registrados se han dividido en tres niveles: Principiante, Aficionado y Experto; explicados ya en el apartado 3.2 de este capítulo. Además, se han fijado unos logros con los que éstos pueden conseguir puntos para subir de nivel. La propuesta para ello se explica en las siguientes secciones.

3.6.1. Niveles, logros y puntos

Inicialmente, todos los usuarios registrados forman parte del nivel Principiante y, a medida que participan en la aplicación, consiguiendo logros, (subiendo anotaciones, editando, creando nuevos árboles, etc.) van ganando puntos. Los puntos que se ganan con cada logro se especifican en la Tabla 3.1. Además, en la Tabla 3.2 se especifican los puntos necesarios para subir de nivel, dependiendo del nivel actual.

El *backend* realizaría gran parte de esta propuesta, ya que sería el encargado de almacenar los puntos de los usuarios en cada momento. El cambio de nivel podría dejarse en manos del *frontend*, pues simplemente debería comparar los puntos con los umbrales de la Tabla 3.2.

Cada nivel estará asociado a un icono en forma de medalla (Oro, Plata, Bronce) respectivamente. Esta medalla es meramente informativa para el resto de usuarios pudiendo ayudar a la credibilidad de los datos añadidos, ya que, por ejemplo, si un principiante ve que un usuario Experto ha validado la información, puede resultarle más convincente.

LOGRO	DESCRIPCIÓN	PUNTOS
Crear un árbol	Un usuario registrado publica información acerca de un árbol nuevo. La información mínima a añadir debe ser la posición del mismo	5
Crear anotación de especie	Un usuario añade la especie a un árbol existente	3
Crear anotación de posición	Un usuario añade una nueva posición de un árbol existente	2
Crear anotación de imagen	Un usuario añade una imagen a un árbol existente	1
Validar anotación	Cualquier usuario registrado que esté utilizando la aplicación puede contribuir en la validación de la información publicada. Para ello se añadirá un pequeño apartado en cada anotación que le permita indicar si está de acuerdo con dicha información	4

Tabla 3.1: Especificación de los puntos conseguidos en función de los logros

Nivel actual	Nivel siguiente	Puntos necesarios
Principiante	Aficionado	35
Aficionado	Experto	100

Tabla 3.2: Especificación de los puntos necesarios para subir de nivel

3.6.2. Validación de datos

La aplicación no dispone de expertos del dominio forestal para dar como verdaderos los datos subidos por los usuarios, pero sí puede categorizar automáticamente a determinada información como la más fiable. Por ello se propone nombrar a las anotaciones más fiables “**Anotaciones primarias**”.

Las anotaciones primarias sirven para informar al usuario de qué anotaciones son las que la mayor parte de los usuarios creen que son correctas. Esta puede ser una manera de solucionar el conflicto que se da en caso de que existan anotaciones contradictorias (por ejemplo, dos anotaciones de especie diferentes). Para que una anotación pase a ser la anotación primaria de un árbol, es necesario que varios usuarios registrados hayan confirmado que la información es correcta. En este punto es donde los niveles de usuario toman relevancia, ya que la confirmación de un usuario de nivel Experto pesa más que la de uno Principiante. En la Tabla 3.3 se indica el peso de cada confirmación, dependiendo del nivel que tenga el usuario que la realiza.

De esta manera, la anotación con más confirmaciones pasará automáticamente a ser la primaria. El *backend* sería el encargado de llevar a cabo este proceso de actualización de

PRINCIPIANTE	AFICIONADO	EXPERTO
1	3	5

Tabla 3.3: Especificación de los pesos para validar una anotación

validaciones y de cálculo de qué anotación es la primaria en cada momento. Por su parte, el *frontend* se encargaría de indicar al usuario quiénes la han validado, además de darle mayor visibilidad con una estrella y el texto de: “**Anotación más popular**”.

3.7. Conclusiones

Con el proceso de análisis se ha comprendido el dominio de aplicación de nuestro software, qué debe hacer el sistema detallando sus funcionalidades y se han analizado algunas de sus restricciones (conectividad, eficiencia, necesidades, etc.). Dicho proceso ha ocupado una gran parte de las iteraciones de la metodología iterativa e incremental utilizada.

Tras completar las fases de arranque y elaboración de la metodología del Proceso Unificado, se pudo pasar a la fase de construcción, que consiste principalmente en el diseño e implementación de la aplicación *Timber*. En ese momento fue relevante determinar qué funcionalidades eran más críticas para el correcto funcionamiento de la aplicación. De esta manera, los esfuerzos en las primeras iteraciones estuvieron centrados principalmente en la parte correspondiente a la gestión de árboles y anotaciones. Una vez alcanzados los objetivos de estas funcionalidades, se comenzó con la parte de gestión de usuarios. El proceso completo se detalla en el capítulo 4.

Capítulo 4

Diseño e implementación

4.1. Introducción

Una vez especificados los requisitos que debe cumplir la aplicación a desarrollar, y antes de empezar con la implementación, se debe especificar la arquitectura que va a seguir el sistema en cuestión, distinguiendo las diferentes partes que lo componen y las comunicaciones necesarias entre ellas. Además, es importante estudiar las diferentes tecnologías que se van a utilizar, atendiendo a sus ventajas, restricciones y tiempo disponible. Todo ello se recopila en este capítulo, que es el resultado de las fases de construcción y transición de la metodología del Proceso Unificado indicada en la sección 1.3 de este documento. La fase de construcción resultó ser la más larga del ciclo de vida del sistema software, pues completa la implementación del análisis obtenido en la fase de elaboración. Esta fase está formada por una serie de iteraciones, donde las tareas predominantes fueron el diseño del sistema y la implementación del mismo. En cada iteración se iban incluyendo nuevas funcionalidades de la aplicación, al final de cada una de las cuales se iba obteniendo una nueva versión ejecutable de *Timber*. La fase de construcción también incluyó la actividad de pruebas, que normalmente consistía en presentar los avances realizados en las reuniones con los tutores del TFG y otros participantes ya mencionados anteriormente. Finalmente, en la fase de transición se llevó a cabo el despliegue de *Timber* en el entorno de producción y las correspondientes pruebas del producto con miembros del GSIC-EMIC en condiciones de laboratorio. Esta fase permitió resolver incidencias en lo relativo a la implementación, así como mejorar ciertos aspectos de la versión presentada gracias a la retroalimentación de los participantes en las pruebas. Esta fase concluyó con la creación de un manual de usuario para facilitar la formación de los usuarios en la aplicación. Dicho manual está disponible en el Apéndice D.

Este capítulo hace referencia al diseño e implementación del *frontend* y se organiza como sigue: en el apartado 4.2 se analizan las diferentes tecnologías utilizadas para el desarrollo del proyecto. En el apartado 4.3 se detalla la arquitectura del sistema. En el apartado 4.4 se explica prototipo concreto que implementa la arquitectura. En el 4.5, se detallan los pasos necesarios para el despliegue de la aplicación en producción. El prototipo final alcanzado se muestra en el apartado 4.6. Finalmente, se concluye este capítulo con algunas conclusiones acerca del mismo.

4.2. Tecnologías utilizadas

Para la implementación de la aplicación *Timber* se hace uso de un conjunto de tecnologías, herramientas y *frameworks* que facilitan el trabajo de desarrollo del sistema software. En este apartado se explican cada una de ellas, además de mostrar los principales motivos por los que se ha optado por su elección.

El primer paso fue elegir el tipo de aplicación a desarrollar. Para ello se estudiaron dos posibilidades descritas en [Cha11]: (i) aplicación nativa o (ii) aplicación web. Una aplicación nativa está desarrollada y optimizada específicamente para un sistema operativo determinado y la plataforma de desarrollo del fabricante (Android, Windows, iOS, etc). Este tipo de aplicaciones se adapta casi al completo con las funcionalidades y características del dispositivo, obteniendo así una experiencia de uso satisfactoria para el usuario final. Por otro lado, una aplicación web no depende de ningún sistema operativo, sino que se adapta a cualquier navegador sea cual sea el sistema operativo. La desventaja de este tipo de aplicaciones es que no funcionan sin conexión a Internet. La aplicación *Timber* fue pensada para utilizarse principalmente en dispositivos móviles (teléfonos y tabletas), facilitando su uso al aire libre, aunque también podría usarse en ordenadores de sobremesa. Es por ello que la compatibilidad multiplataforma fue el requisito clave para optar por una **aplicación web**, ya que el camino para la generación de código compatible es mucho más sencillo, pues no depende del sistema operativo en el que se utilice. Otro argumento a favor de las aplicaciones web es su facilidad de instalación, pues suele ser transparente para el usuario final.

Una vez optado por la creación de una aplicación web, se indagó en las ventajas que supondría implementar una **SPA**¹ (*Single Page Application* o aplicación de una página). Una SPA es una aplicación web que carga al completo todos los recursos de la aplicación en la solicitud inicial. Está compuesta por componentes individuales que pueden ser reemplazados/actualizados de manera independiente, de modo que no es necesario recargar toda la página en cada acción del usuario. Una ventaja que presentan estas aplicaciones con respecto a las tradicionales es que las SPA no están fuertemente ligadas al *backend*, como lo están las otras [Jad15]. Este tipo de aplicaciones ya no depende de un *backend* que genere las páginas, pues la propia aplicación es capaz de generar todo el contenido sin necesidad de realizar ninguna llamada a la parte servidora. Esta separación de *frontend* y *backend* es la que ha permitido que la aplicación *Timber* pueda haberse desarrollado de forma independiente y paralela con Cristina Mayo.

A continuación, se explican en detalle las tecnologías más importantes utilizadas en el proyecto. Están divididas en dos grupos: el primero de ellos (4.2.1) corresponde con las tecnologías propias del desarrollo del software y el segundo (4.2.2) con las tecnologías manejadas para la gestión del proyecto y el despliegue en producción.

¹https://es.wikipedia.org/wiki/Singlepage_application

4.2.1. Tecnologías de desarrollo software

Angular

Angular² es un marco de diseño de aplicaciones (o *framework*) de código abierto y una plataforma de desarrollo para crear aplicaciones eficientes y sofisticadas de una sola página (SPA). Está programado en TypeScript y es mantenido por Google. **TypeScript**³ es un lenguaje de código abierto que se basa en JavaScript, una de los lenguajes de programación más utilizados en el mundo, añadiendo definiciones de tipos estáticos. La declaración de tipos proporciona una forma de describir al objeto que permite a TypeScript una mejor validación del código.

Angular es conocido por ser un *framework* orientado a la creación de aplicaciones web con una curva de aprendizaje bastante corta comparado con otros, permitiendo centrarse en el desarrollo de la aplicación más que en el aprendizaje del *framework*. Este punto a favor fue decisivo para su elección. Además, para usarlo basta con conocer **JavaScript**, **HTML** (*HyperText Markup Language*) y **CSS** (*Cascading Stylesheets*). Todas ellas herramientas de desarrollo web destinadas a la parte del cliente, con las que la autora de este TFG ya tenía cierta familiaridad.

La arquitectura de una aplicación Angular se basa en ciertos conceptos fundamentales, que son:

- Los bloques de construcción básicos son los **módulos** o **NgModules**, que proporcionan un contexto de compilación para los componentes. Los **NgModules** recogen código que se relaciona según conjuntos funcionales. Una aplicación siempre tiene al menos un módulo raíz, que permite el arranque (*bootstrapping*), y normalmente tiene muchos más módulos de características.
- Los **componentes** se encargan de definir las vistas, que son conjuntos de elementos de pantalla que Angular puede elegir y modificar según la lógica y los datos de su programa. Estos componentes son clases que tienen **propiedades** y **métodos**. Los metadatos de cada componente se asocian a una **plantilla** (o *template*, encargada de definir la vista). Una plantilla combina el HTML ordinario con las directivas de Angular y el marcado (CSS) que permite a Angular modificar el HTML antes de mostrar su visualización.
- Los componentes utilizan **servicios**, que proporcionan una funcionalidad específica no relacionada directamente con las vistas. Los proveedores de servicios pueden ser inyectados en los componentes como dependencias, haciendo que su código sea modular, reutilizable y eficiente. Se utilizan los servicios de Angular, por ejemplo, para las tareas de solicitud y envío de datos con el servidor.

Angular CLI

Angular CLI⁴ (*Command Line Interface*) es un intérprete de línea de comandos diseñado por el equipo de Angular que facilita la creación y el desarrollo de proyectos con este *framework*. Al crear un nuevo proyecto, crea la estructura necesaria para el funcionamiento de la aplicación y provee de un conjunto de comandos muy útiles para acelerar la creación

²<https://angular.io/>

³<https://www.typescriptlang.org/>

⁴<https://cli.angular.io/>

de código, automatizando tareas como la creación de módulos, servicios y componentes. También facilita la creación de los archivos necesarios en la etapa de producción.

Angular CLI es una herramienta **NodeJS**⁵, por lo que es necesario tenerlo instalado en nuestro sistema operativo. Además, se instala con el gestor de paquetes **NPM**⁶ (Node Package Manager), sistema multiplataforma que utiliza NodeJS por defecto.

Bootstrap

Bootstrap⁷ es un *framework* **CSS** que permite dar forma y estilo a un sitio web mediante librerías CSS predefinidas (tipografías, cuadros, botones, menús, etc).

Bootstrap ha sido utilizado en este proyecto para crear el diseño de la interfaz de usuario, pues proporciona las herramientas necesarias para crear interfaces limpias y totalmente adaptables a cualquier tipo de dispositivo y pantalla de una manera rápida y eficaz.

4.2.2. Otras tecnologías

GitHub

GitHub⁸ es el sistema de gestión de proyectos y control de versiones de código para desarrolladores por excelencia. Es también uno de los repositorios online más grandes de trabajo colaborativo en todo el mundo⁹.

GitHub se ha utilizado durante la implementación de la aplicación *Timber* para llevar el control de las versiones con **Git**, pero además para sincronizar y guardar en un repositorio el trabajo conjuntamente de *frontend* y *backend*. El repositorio en el que se encuentra el proyecto completo está accesible mediante el siguiente enlace:

<https://github.com/gsic-emic/SocialTreeAnnotation>

Docker

Docker¹⁰ es un proyecto de código abierto que es utilizado para automatizar el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Docker ha sido utilizado para realizar el despliegue de la aplicación y unificar *frontend* y *backend* en un mismo contenedor.

Nginx

Nginx¹¹ es un software de servidor web de código abierto que sirve como proxy inverso, balanceador de carga HTTP y proxy de correo electrónico para IMAP, POP3 y SMTP. Ofrece una arquitectura asíncrona y controlada por eventos, característica que le hace ser uno de los servidores más confiables para la velocidad y la escalabilidad. Nginx ha sido escogido como servidor web para la aplicación *Timber* debido a su velocidad y excelente capacidad para manejar muchas conexiones a la vez.

⁵<https://nodejs.org/es/>

⁶<https://www.npmjs.com/>

⁷<https://getbootstrap.com/>

⁸<https://github.com/>

⁹<https://octoverse.github.com/>

¹⁰<https://www.docker.com/>

¹¹<https://www.nginx.com/>

4.3. Arquitectura de la aplicación

En la Figura 4.1 se plantea una representación en caja negra de la arquitectura de la aplicación web *Timber* en su totalidad. Esta arquitectura multicapa está formada por *frontend* y *backend*. La arquitectura propia del *frontend* es la que se especifica en esta sección, correspondiente al presente TFG. La arquitectura del *backend* se puede consultar en [May20]. En la Figura 4.1 también se observa que el conjunto de *frontend* más *backend* conforman la aplicación web *Timber* y que ésta hace uso del servidor web Nginx.

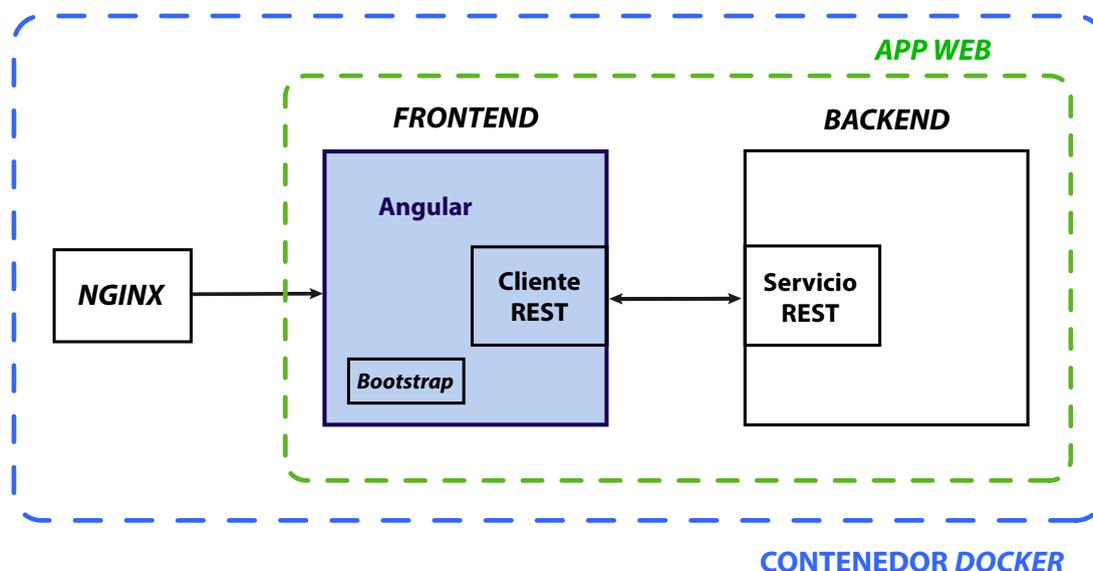


Figura 4.1: Arquitectura cliente-servidor con API REST

Frontend y *backend* se comunican mediante una **API** (*Application Programming Interface*) con arquitectura **REST** (*REpresentational State Transfer*), diseñada en [May20]. De ese modo, las comunicaciones entre ambos son sin estado (*stateless*). Esto facilita, por un lado, su implementación y, por otro, permite descargar a la parte servidora de carga de trabajo, ya que es el *frontend* el que se encarga de mantener el estado global de la aplicación, siendo el *backend* el responsable de mantener el estado de los recursos [Fie00]. Además, esta estructura permite utilizar la API diseñada en otras aplicaciones que hagan uso de los mismos datos. El *frontend* consume los datos ofrecidos por la API, por lo que dentro de este se especifica un cliente REST (ver Figura 4.1), que es el encargado de realizar las peticiones HTTP asíncronas de los recursos. La API REST proporciona dichos recursos en formato JSON (*JavaScript Object Notation*). Los recursos expuestos por la API se especifican en la Tabla 4.3.

Tras describir la manera en la que el *frontend* se comunica con el *backend* para obtener los datos necesarios de la aplicación, pasamos a detallar más en profundidad la arquitectura diseñada en este TFG (la del *frontend*). La estructura viene determinada por la propia de Angular. En la Figura 4.2 se puede observar de manera esquemática cómo se relacionan las diferentes piezas de Angular correspondientes a un módulo, todas ellas definidas en el apartado 4.2 de este documento.

La aplicación está compuesta por varios **módulos**. Estos módulos están formados por

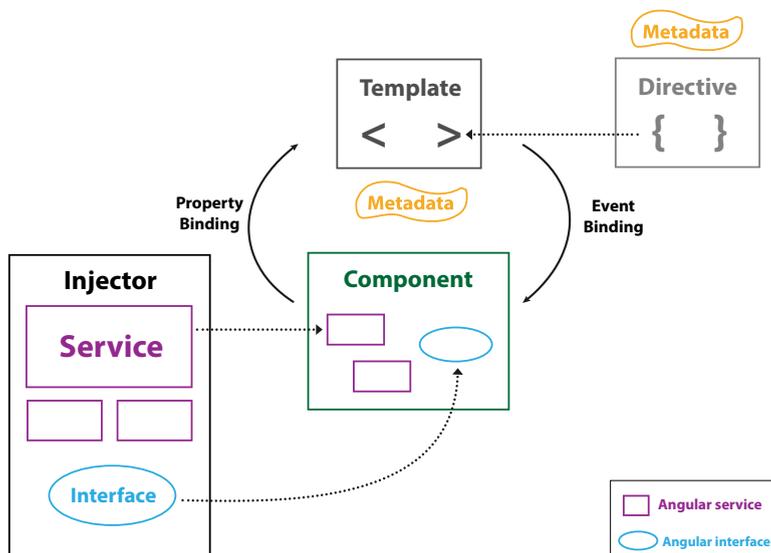


Figura 4.2: Arquitectura de un módulo de Angular

una **plantilla** (*template*) en HTML y etiquetas personalizadas de Angular (llamadas **directivas**), que representa una parte de la interfaz, y un **componente**, que es el encargado de controlar la lógica de la plantilla y los datos asociados a la misma. Los componentes son exportados como clases, con propiedades y métodos. Las directivas de una plantilla poseen lógica de programación, enlazando la lógica de los componentes (**data binding**) con las vistas de la misma. Como se observa en la Figura 4.2, hay dos tipos de *data binding*:

- **Event binding** o enlace de eventos. Se dan cuando el usuario interactúa con la aplicación modificando alguna entrada, de manera que se actualizan los datos del componente.
- **Property binding** o enlace de propiedades. El componente modifica el valor de alguno de los datos y el resultado se refleja en el HTML.

Dentro de la aplicación de Angular se han definido diferentes componentes para cada una de las vistas de la interfaz de usuario (vista principal, vista de la página de registro, vistas de las páginas de usuarios registrados, etc). Además, muchas de las vistas están formadas por varios componentes. Algunos de ellos se usan simplemente para dar estilo a las vistas, utilizando clases CSS predefinidas con Bootstrap, mientras que otros se encargan de la visualización de los datos procedentes del *backend*. En la Tabla 4.1 se listan todos los recursos expuestos por la aplicación, así como los componentes que lo forman.

Los **servicios** también están compuestos por lógica para manejar datos de la aplicación y estructuras de datos predefinidas en forma de **interfaces**. Tanto servicios como interfaces son utilizados por los componentes. La aplicación *Timber* está compuesta por una serie de **servicios** encargados cada uno de un conjunto de tareas concretas. Su función principal es definir una serie de métodos que luego serán utilizados en los diferentes componentes de la aplicación. En la Figura 4.3 se puede ver una representación de los mismos, junto con sus interfaces asociadas. Los servicios se explican a continuación:

- **API SERVICE**. Servicio encargado de la comunicación con la API REST del *backend*. El resto de servicios hacen uso de sus métodos, pues este servicio se encarga

Recurso	URL	Componentes
Inicio	https://timber.gsic.uva.es/	InicioComponent, MenuComponent, MapaComponent, ListaComponent
Sobre Nosotros	https://timber.gsic.uva.es/contacto	MenuComponent, ContactoComponent
Registrarse	https://timber.gsic.uva.es/registro	MenuComponent, RegistroComponent
Iniciar Sesión	https://timber.gsic.uva.es/inicio_sesion	MenuComponent, IniciarSesionComponent
Principal registrados	https://timber.gsic.uva.es/principal	MenuUsuariosComponent, PrincipalComponent, MapaComponent, ListaComponent
Nuevo Árbol	https://timber.gsic.uva.es/nuevo_arbol	MenuUsuariosComponent, NuevoArbolComponent
Mis árboles y anotaciones	https://timber.gsic.uva.es/mis_anot	MenuUsuariosComponent, MisAnotacionesComponent, MilistaComponent
Buscador	https://timber.gsic.uva.es/busqueda	MenuUsuariosComponent, BusquedaComponent
Comunidad	https://timber.gsic.uva.es/comunidad	MenuUsuariosComponent, ComunidadComponent
Ajustes	https://timber.gsic.uva.es/ajustes	MenuUsuariosComponent, AjustesComponent

Tabla 4.1: Recursos ofrecidos por el *frontend* de *Timber*

tanto de recuperar como de actualizar datos relacionados con los árboles, las anotaciones y las imágenes, provenientes del *backend*. El servicio *API SERVICE* está pensado para apoyar todos los casos de uso de gestión de árboles y anotaciones y los de navegación y búsqueda.

- **TREE SERVICE.** Se encarga de funciones relacionadas con la gestión y adaptación de los datos procedentes de los árboles de la aplicación. Hace uso de la interfaz *Tree*, que especifica las propiedades que posee el objeto de tipo árbol. El servicio *TREE* está pensado para apoyar los casos de uso *RegistrarNuevoArbol*, *ConsultarMisArboles*, *ConsultarArbolesZona*, *ConsultarArbolCompleto* y *FiltrarArboles* definidos en la sección 3.4.
- **SPECIES SERVICE.** Servicio que se encarga de la gestión de los datos procedentes de las especies recuperadas del *backend*. Estas especies siguen el modelo RDF del *Cross-Forest*, por lo que necesitan ser extraídas y expuestas a los usuarios de manera entendible para ellos. Se ayuda de la interfaz *Species* para la adaptación

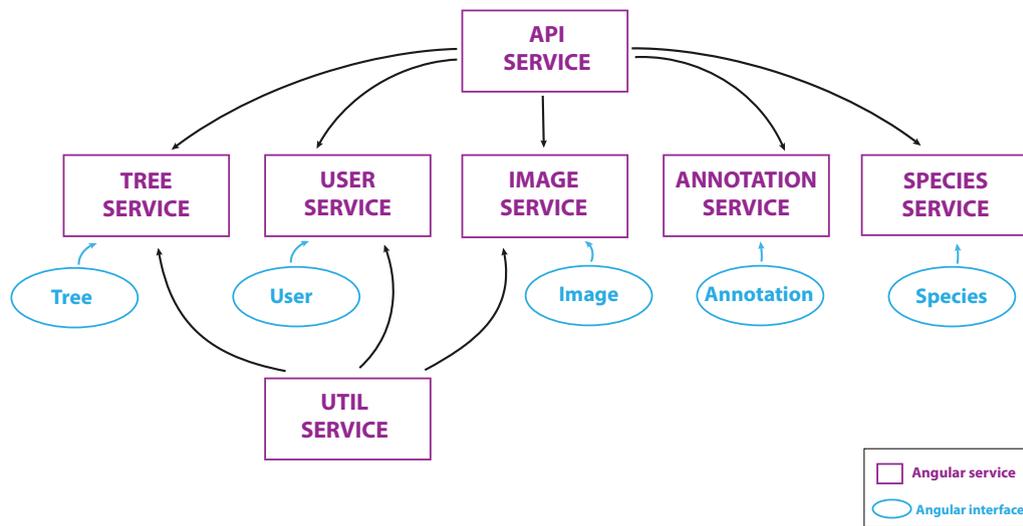


Figura 4.3: Servicios de la arquitectura de la aplicación

de los datos, pues dicha interfaz representa las características típicas de la especie de un árbol. El servicio *SPECIES* está pensado para apoyar todos los casos de uso de gestión de árboles y anotaciones y los de navegación y búsqueda que muestren al usuario la especies de cada árbol.

- **ANNOTATION SERVICE.** Encargado de la gestión de los datos que forman las anotaciones. Hace uso de la interfaz *Annotation*, la cual especifica todas las propiedades de las anotaciones. El servicio *ANNOTATION* está pensado para apoyar los casos de uso *AñadirAnotacion*, *ConsultarMisAnotaciones*, *ValidarAnotacion*, *EditarAnotacion*, *EliminarAnotacion* y *ConsultarArbolCompleto* definidos en la sección 3.4.
- **IMAGE SERVICE.** Dentro de los tipos de anotaciones, las de imagen requieren algo más de trabajo que el resto. Es por ello que se ha creado un servicio especial encargado únicamente de la gestión, creación y adaptación de las imágenes, así como la especificación del título, parte que representa y descripción. Todas estas características se encuentran declaradas en la interfaz *Image*. El servicio *IMAGE* está pensado para apoyar los casos de uso *AñadirAnotacionImagen* y *ConsultarArbolCompleto* definidos en la sección 3.4.
- **USER SERVICE.** Se encarga de funciones relacionadas con la gestión y adaptación de los datos procedentes de los usuarios de la aplicación. Hace uso de la interfaz *User*, que especifica las propiedades que posee el objeto de tipo usuario. El servicio *USER* está pensado para apoyar todos los casos de uso del grupo de gestión de usuarios definidos en la sección 3.4.1.
- **UTIL SERVICE.** Servicio que recoge funcionalidades de uso común en varios componentes, como por ejemplo formatear la fecha con una determinada expresión, crear mensajes de error genéricos, obtener datos de la sesión, etc.

Las interfaces mencionadas anteriormente son las encargadas de modelar los datos intercambiados con la API del *backend*. El formato utilizado para el intercambio es JSON. Estas interfaces están compuestas por los atributos necesarios para definir cada uno de

los objetos que representan. En la Tabla 4.2 podemos observar las diferentes interfaces declaradas en la aplicación, junto con los atributos de las mismas.

Interfaz	Atributos
Tree	identificador, latitud, longitud, especie, creador, fecha
Annotation	identificador, creador, fecha, primaria, tipo
Image	identificador, jpegBASE64, titulo, fecha, descripcion, parte
User	nombre, username, email, password, fechaCreacion

Tabla 4.2: Modelado de datos mediante las interfaces

Cuando se realiza la recogida de datos introducidos por el usuario, los datos son almacenados según los modelos de las interfaces especificados en la Tabla 4.2 y convertidos a JSON para ser mandados al *backend* a través de la API. En este caso, no hay una relación directa entre los datos guardados en el *backend* según la ontología y los datos en formato JSON que se mandan a la API. La conversión se hace directamente en el *backend*. Sin embargo, cuando se solicitan datos al *backend*, la API los devuelve en un JSON cuya estructura sigue la de la ontología. En este caso, es el *frontend* el que se encarga de recoger el campo que contiene la información a mostrar (*value*), y almacenarla según las interfaces diseñadas en la Tabla 4.2. Se plantea como posible trabajo futuro realizar esta serialización de los datos a partir de la ontología para que sea más robusto.

4.4. Implementación

La implementación de esta primera versión de la aplicación no cumple con todas las especificaciones detalladas en el capítulo de análisis, de modo que centra sus esfuerzos en las funcionalidades relacionadas con los árboles, dejando a un lado el tema social. El motivo de esta decisión es que la primera versión creada es una prueba de concepto, en la que se quiere constatar que la aplicación permite anotar y visualizar árboles a un usuario individual, requisito previo antes de poner en marcha las funciones sociales. Además, la implementación del *frontend* ha necesitado ir siempre a la par que la del *backend*, pues requiere de muchas funcionalidades del mismo.

Se realizaron tres iteraciones en esta fase de implementación que se dividen según los casos de uso implementados. En primer lugar, se dio importancia a la visualización de las características de los árboles previamente almacenados en el *backend* (gracias al conjunto de Datos Abiertos del IFN) así como a su localización, correspondiéndose con los casos de uso *ConsultarArbolesZona* y *ConsultarArbolCompleto*. Para el caso de uso *ConsultarArbolesZona* se decidió la inclusión de un mapa que facilitase consultar la localización de todos los árboles registrados. Los detalles de esta implementación se pueden ver más adelante en este mismo apartado. En la segunda iteración, se pasó a implementar los casos de uso del grupo de gestión de árboles y anotaciones, puesto que se consideró que la creación de nuevos árboles y anotaciones en la aplicación eran requisitos imprescindibles para esta

primera versión. Los casos de uso implementados fueron: *RegistrarNuevoArbol*, *AñadirAnotacionEspecie*, *AñadirAnotacionLocalizacion* y *AñadirAnotacionImagen*. Sin embargo, estos últimos necesitan de un usuario registrado para realizarse, por lo que los casos de uso *DarseDeAlta*, *IniciarSesion* y *CerrarSesion* tuvieron también lugar en esta segunda iteración. En la tercera iteración se implementaron otros casos de uso que requerían la existencia de usuarios registrados como *ConsultarMisDatos*, *ConsultarMisArboles*, *ConsultarMisAnotaciones* y *FiltrarArboles*.

Los casos de uso no implementados en esta primera versión son todos los relacionados con modificar o eliminar datos (de árboles, anotaciones y usuarios) que requieren la acción del *backend* para ello. Además, las funcionalidades propias de los niveles de usuarios y curado de datos explicadas en el apartado 3.6 quedan también pendientes.

4.4.1. Estructura del proyecto

Gracias a la herramienta **Angular CLI**, crear el proyecto de la aplicación es muy sencillo. Angular CLI ofrece una serie de comandos que nos facilitan la creación de estructura de ficheros necesaria para el correcto funcionamiento del proyecto. Además, de comandos para crear módulos, servicios, componentes, etc. La estructura de carpetas se especifican a continuación:

- **e2e:** esta carpeta, denominada “*end to end*”, engloba una serie de ficheros cuya función es la realización de test automáticos, como si un usuario interactuase con la aplicación. Se ejecuta con el comando "ng e2e".
- **node_modules:** carpeta que contiene todas las dependencias del proyecto.
- **src:** directorio de trabajo. En él se crean los diferentes módulos, por lo que contiene todo el código programado de la aplicación. Lo forman varios ficheros y carpetas:
 - Carpeta **app:** en esta carpeta se ubica la implementación de todos los módulos. Está formada por tantas carpetas como módulos tiene el proyecto. Dentro de cada una de ellas se encuentra el componente del módulo asociado, su HTML y su archivo de estilo CSS.
 - Carpeta **assets:** carpeta donde se almacenan los recursos, imágenes y archivos necesarios para que el proyecto funcione.
 - Carpeta **environments:** en ella se encuentran las configuraciones y variables de entorno necesarias para lanzar el proyecto tanto en desarrollo como en producción.
 - **favicon.ico:** archivo del icono del proyecto.
 - **index.html:** página principal del proyecto.
 - **main.ts:** archivo Type Script inicial del proyecto donde se declaran todas las configuraciones globales.
 - **editorconfig:** archivo que contiene la configuración del editor de código.
 - **.gitignore:** especifica las carpetas o archivos que debe ignorar el *git* cuando se añade el proyecto al repositorio.
 - **angular.json:** contiene la configuración de Angular. Además, incluye rutas, versiones, etc.

- ***package.json***: archivo con la configuración de la aplicación. Contiene el nombre de la app, las dependencias necesarias para su correcta ejecución y muchas otras cosas.
- ***tsconfig.json***: contiene la configuración TypeScript.
- ***tslint.json***: archivo utilizado para que el código sea sostenible y se mantenga.

4.4.2. Detalles de implementación

El proyecto está programado en TypeScript. Este lenguaje es fácilmente entendible por cualquier programador experimentado en JavaScript, con la diferencia de que las diferentes variables tienen especificado el tipo de dato que representan. En esta sección se explican con detalle algunas de las características más relevantes de la implementación de la aplicación.

Comunicación con el *backend*

Una de las partes más relevantes dentro de la implementación del *frontend* es la comunicación con el *backend* para el intercambio de datos con el almacén de triplas. Como ya se ha explicado anteriormente, se cuenta con una API REST que facilita tal comunicación y que ofrece los recursos especificados en la Tabla 4.3.

En la Figura 4.1 se hablaba de un cliente REST que era el encargado de establecer la comunicación. Este cliente está implementado en Angular mediante los servicios *API SERVICE* y *USER SERVICE* descritos en el apartado anterior. Para realizar las consultas HTTP, dichos servicios hacen uso de una API proporcionada por Angular denominada `HttpClient`¹². Los métodos ofrecidos por esta API realizan operaciones de comunicación asíncronas, permitiendo que el flujo de la aplicación no se quede bloqueado. Para la implementación de estas operaciones, Angular utiliza **programación reactiva** con la extensión `RxJS`¹³ (*Reactive JavaScript*), que es una librería de programación reactiva de JavaScript para trabajar con *streams* de datos mediante el uso de observables. Un observable consiste en una representación de lo que va devolviendo un método asíncrono a lo largo del tiempo.

Las peticiones se realizan mediante **métodos de petición**, que son los encargados de indicar la acción que se desea realizar a un recurso determinado. Los métodos utilizados en esta versión de la aplicación son:

- **GET**: método que se utiliza para recuperar datos. Estos datos corresponden con la información de los árboles, de las especies, de las anotaciones y de los usuarios disponible en el almacén de triplas. Por ejemplo, si el cliente quiere mostrar al usuario todos los árboles guardados en el sistema, realizaría una petición HTTP GET al recurso Árboles: `https://timber.gsic.uva.es/sta/data/tree` (consultar Tabla 4.3), obteniendo un JSON con la respuesta (Ver Figura 4.4). El identificador de cada árbol corresponde tanto a la URL de dicho recurso (dónde se encuentra), como a la URI que representa dicho árbol dentro del lenguaje RDF.

Si ahora quisiéramos recuperar, por ejemplo, toda la información del árbol seleccionado en el ejemplo anterior, bastaría con utilizar el método GET y la URL del identificador de dicho árbol: `http://timber.gsic.uva.es/sta/data/tree/20200621-a3660` (recurso Árbol de la Tabla 4.3). En la Figura 4.5 se muestra el resultado obtenido.

¹²<https://angular.io/api/common/http/HttpClient>

¹³<https://rxjs-dev.firebaseapp.com/>

Recurso	URL	Datos ofrecidos
Raíz	<code>https://timber.gsic.uva.es/sta/</code>	Especifica el resto de recursos ofrecidos
Usuarios	URL raíz + <code>data/user/</code>	Todos los usuarios dados de alta en la aplicación así como su información
Usuario	URL raíz + <code>data/user/:userId</code>	Datos personales del usuario con <i>nickname</i> <code>userId</code>
Árboles	URL raíz + <code>data/tree</code>	Todos los árboles almacenados en el <i>backend</i>
Árbol	URL raíz + <code>data/tree/:treeId</code>	Información almacenada del árbol cuyo identificador es <code>treeId</code>
Anotaciones	URL raíz + <code>data/annotation</code>	Todas las anotaciones almacenados en el <i>backend</i>
Anotación	URL raíz + <code>data/tree/:annotationId</code>	Información almacenada de la anotación cuyo identificador es <code>annotationId</code>
Imágenes	URL raíz + <code>data/image</code>	Todas las anotaciones almacenados en el <i>backend</i>
Imagen	URL raíz + <code>data/tree/:imageId</code>	Información almacenada de la imagen cuyo identificador es <code>imageId</code>
Especies	URL raíz + <code>data/species</code>	Todas las especies guardadas en el <i>backend</i>
Parte Árbol	URL raíz + <code>data/treePart</code>	Partes que puede representar la foto de un árbol

Tabla 4.3: Recursos ofrecidos por la API REST del *backend*

Además, podemos observar en esta figura que los datos devueltos en el JSON siguen una estructura particular, respondiendo a los diferentes lenguajes RDF explicados en el Capítulo 2 de este documento y a la ontología propia de la aplicación detallada en el Anexo A.

Una vez obtenidos los datos deseados, el *frontend* se encarga de adaptarlos para su correcta visualización en la interfaz de usuario.

- **POST**: método que se utiliza para pedir la creación de un recurso (Árbol, Anotación, Usuario). En la aplicación se usa HTTP POST con **autorización básica** [Res15] para crear tanto árboles, como anotaciones y usuarios. Esta autenticación se realiza añadiendo al método POST una cabecera de autorización que sigue la forma `usuario:contraseña`.

La implementación de otros métodos como PUT (editar) o DELETE (borrar) no se

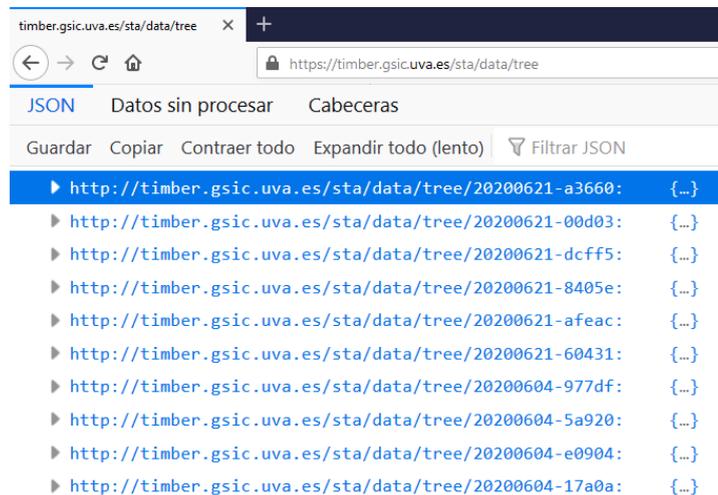


Figura 4.4: Resultado obtenido con el método GET al recurso **Árboles**

ha realizado en esta primera versión de la aplicación, por lo que, por el momento, no es posible editar ni eliminar los datos que se añaden en el sistema.

Mapa

El interés principal de la aplicación reside en la interacción con un mapa en el que se muestran todos los árboles almacenados en el sistema. La implementación de dicho mapa se ha realizado con *Leaflet*¹⁴, una librería de JavaScript de código abierto utilizada para la publicación e interacción de mapas en la Web y fácilmente adaptables para dispositivos móviles. Además, *Leaflet* proporciona una API fácil de usar.

En la aplicación se ha creado un componente dedicado exclusivamente a la creación del mapa, el *MapaComponent*. En dicho componente se crea el objeto *mymap*, al que se le añade una capa con información geográfica (denominada *Tile Layer*), además de *popups* con iconos diseñados en forma de árboles, que indican la posición de cada uno de los árboles registrados en la aplicación. La capa del mapa está extraída de la API de *OpenStreetMap*¹⁵, un proyecto colaborativo para crear mapas editables y libres. El mapa incluye adicionalmente un *plugin* denominado *Leaflet.Locate*, que facilita la adición de un botón dentro del mapa para geolocalizar al usuario.

4.5. Despliegue

Durante la **etapa de desarrollo** de la aplicación, el código se va probando en un servidor web local que ejecuta Angular CLI (en el puerto 4200 por defecto) con el comando "ng serve -open". El comando se ejecuta en la carpeta donde se encuentra la aplicación. Al disponer de la opción - -open se abre automáticamente el navegador web una vez compilado el código y, en cuanto se detectan cambios en el código, vuelve a compilarse y se actualiza la vista en el navegador.

¹⁴<https://leafletjs.com/index.html>

¹⁵<https://es.wikipedia.org/wiki/OpenStreetMap>

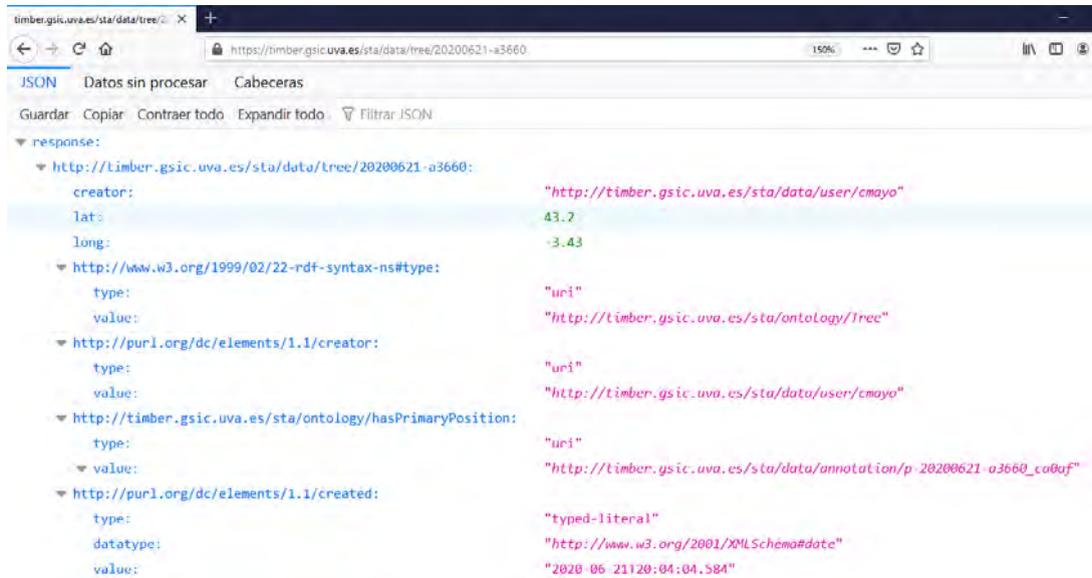


Figura 4.5: Resultado obtenido con el método GET al recurso **Árbol**

Una vez que la aplicación está completamente terminada, se pasa a la **etapa de producción**, en la cual se despliega la aplicación en un servidor remoto. Para ello se usa el comando `"ng build - --prod"`. Este comando compila la aplicación Angular, genera los archivos necesarios para pasar a producción y los guarda en una carpeta de salida denominada *dist*. Tanto `ng build` como `ng serve` limpian la carpeta donde se generan los ficheros de salida antes de compilar el proyecto, pero solo el comando `ng build` escribe los artefactos de compilación generados en la carpeta de salida. Además, este comando reduce el tamaño del código, quitando los espacios en blanco y acortando el nombre de las variables. También precompila el código de forma que el cliente no tenga que bajarse el compilador cuando visita la web.

El despliegue de la aplicación *Timber* se ha realizado en el mismo servidor web en el que se encuentra alojado el *backend* de la misma, es decir, en un servidor **Nginx**. Este servidor se ejecuta sobre una máquina virtual en uno de los ordenadores del grupo de investigación GSIC/EMIC de la UVA. Los pasos necesarios para que la aplicación funcione en producción son los siguientes:

- a. Una vez compilado el proyecto, se debe copiar la carpeta de salida *dist* en `/usr/share/nginx/html`, que es la ruta de la carpeta donde Nginx lee los ficheros estáticos.
- b. A continuación, se debe modificar la configuración que tiene Nginx por defecto para que Angular pueda funcionar correctamente. Para ello es necesario sustituir el archivo del servidor `/etc/nginx/nginx.conf` por otro que contenga las siguientes líneas de código:

```

1 server {
2     listen 80;
3     location / {
4         root /usr/share/nginx/html;
5         index index.html index.htm;
6         try_files $uri $uri/ /index.html = 404;

```

```

7   }
8   }

```

Con estos pasos, la aplicación ya está lista para ser utilizada en producción.

Con el objetivo de que la aplicación pueda ser lanzada desde otras máquinas, se ha creado adicionalmente un fichero `Dockerfile` donde se describen los pasos que debe ejecutar el motor Docker para la creación de una imagen capaz de lanzar la aplicación sobre un servidor Nginx. Dicha imagen puede ejecutarse en un contenedor Docker. Para construirla, es necesario ejecutar el comando: `"docker build . -t timber:latest"`, donde `timber` es el nombre que se ha dado a la imagen y el punto se refiere al `Dockerfile` de la carpeta en la que se encuentra el proyecto. Una vez construida la imagen, bastaría con ejecutar `"docker run -d -p 80:80 timber:latest"` para desplegarla en un contenedor de Docker y poder navegar por la aplicación. El documento `Dockerfile` está disponible en el repositorio de GitHub y en el Anexo C del presente documento.

4.6. Prototipo final y pruebas

Dentro de la fase de transición, y tras el despliegue de *Timber* en producción, se realizaron un serie de pruebas con usuarios reales para probar la aplicación. Estos usuarios reales fueron los tutores del TFG, así como otros miembros del departamento GSIC-EMIC. Las pruebas se realizaron sobre el prototipo final, del que se muestran (en las siguientes subsecciones) algunas capturas de cómo queda la interfaz de usuario para algunos de los casos de uso principales. Además, estas capturas de pantalla se han realizado con resoluciones de dispositivos diferentes para poder comprobar que la interfaz se adapta de manera adecuada a los tamaños de pantallas.

En el Apéndice D se explican las acciones básicas que se pueden realizar en la aplicación dentro de este prototipo final. *Timber* está accesible en la URL:

<https://timber.gsic.uva.es/>

Caso de uso: *DarseDeAlta*

En la Figura 4.6 se muestra la interfaz de usuario para crear una cuenta en *Timber*. En ella se muestra un formulario, donde el usuario debe introducir sus datos en los campos: Nombre y Apellidos, Nombre de Usuario, correo electrónico y contraseña. Además, se pide al usuario que escriba su correo electrónico dos veces y repita su contraseña para confirmar que no hay errores de tecleo.

Caso de uso: *ConsultarMisArboles*

En la Figura 4.7 se muestra la interfaz de usuario para el caso de uso *ConsultarMisArboles*. En ella se muestra en formato de lista todos los árboles que el usuario con la sesión iniciada ha creado desde que se dio de alta en la aplicación. En cada árbol se muestra la especie (si se ha especificado), su localización y el nombre del creador. Se observa además que en la parte superior de cada árbol existen dos iconos para editar y borrar el árbol, acciones que no están implementadas para esta versión de *Timber*. Adicionalmente, existe un botón de “Ver más”, que permitiría realizar el caso de uso *ConsultarArbolCompleto*, y el botón “Completar”, para el caso de uso *AñadirAnotacion*.

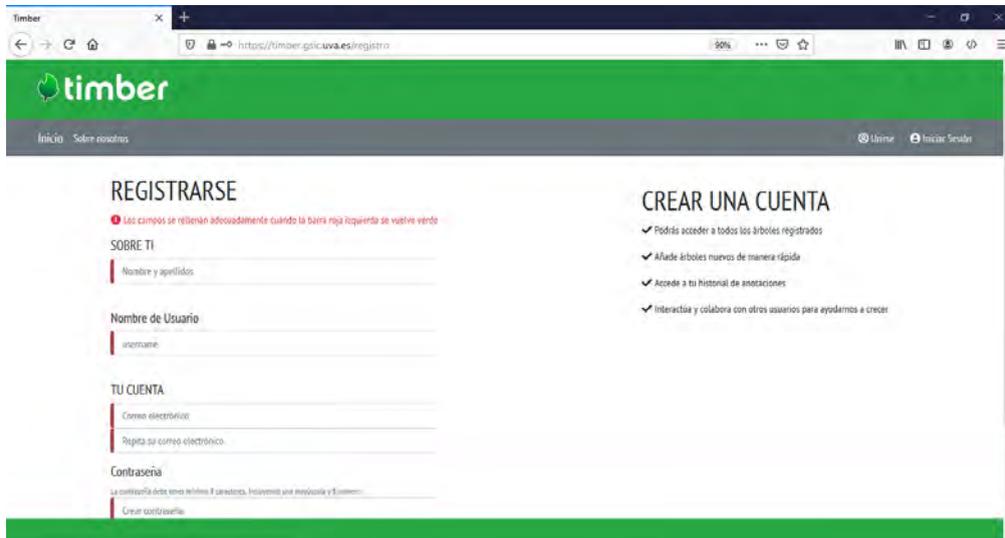


Figura 4.6: Vista de la interfaz para el caso de uso *DarseDeAlta*

Caso de uso: *ConsultarArbolCompleto*

En la Figura 4.8 se muestran varias capturas de la interfaz de usuario para el caso de uso *ConsultarArbolCompleto*. En la captura (a) se muestra la información que la aplicación está tomando como verdadera (especie y localización), además del creador del árbol, la fecha de creación y las fotos (si existen) del mismo. En la captura (b) se muestran las anotaciones asociadas a dicho árbol, donde se observa que distintos usuarios han completado la información del árbol con diferentes anotaciones. Además, la interfaz indica qué anotaciones son las primarias con un icono de una estrella y el título de “Anotación más popular”. También se observa en la esquina inferior izquierda un recuadro verde con el icono “+”, que permite acceder al caso de uso *AñadirAnotacion*.

Se incluye la captura (c) ya que corresponde a un árbol procedente del conjunto de datos del IFN. Estos árboles, a diferencia de los creados por usuarios de la aplicación, muestran información verdadera. Es por ello que la información expuesta está marcada en la interfaz con el texto de “La especie/localización de este árbol ha sido validada por expertos”. La información de este conjunto de árboles podría completarse con imágenes subidas por los usuarios.

Caso de uso: *RegistrarNuevoArbol*

En la Figura 4.9 se muestran un conjunto de capturas de pantalla que representan los pasos básicos para la creación de un nuevo árbol en la aplicación. Las capturas se han realizado fijando la resolución propia de un iPhone 8. En la captura (a) se muestra el principio del formulario, donde el usuario debe introducir las coordenadas del árbol en cuestión o pinchar en el botón “Usar mi ubicación actual”, que hará que dichas coordenadas las coja el dispositivo automáticamente (para ello es necesario permitir la localización en el navegador). En esta captura también se puede seleccionar la especie, con un desplegable que muestra todas las especies almacenadas en el *backend*. Si se desliza hacia abajo, se llega a la captura (b), correspondiente al formulario para subir una imagen del árbol en cuestión. Si se pulsa el botón “Siguiente”, se llega a la captura (c), en la que la interfaz

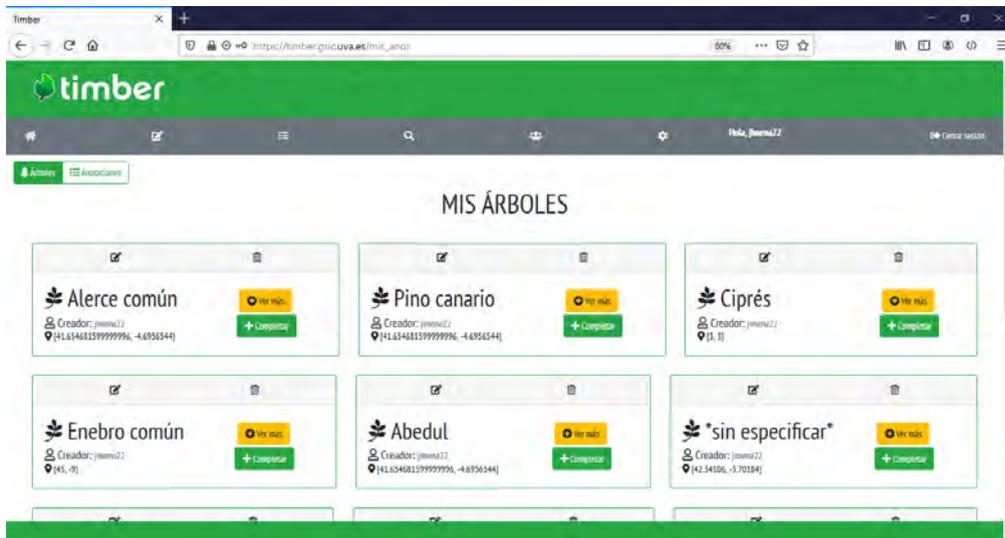


Figura 4.7: Vista de la interfaz para el caso de uso *ConsultarMisArboles*

muestra una previsualización de los datos que se van a mandar al *backend* para la creación del árbol. El caso de uso termina tras pulsar el botón “Confirmar”, llevando al usuario a la captura (d).

4.7. Conclusiones

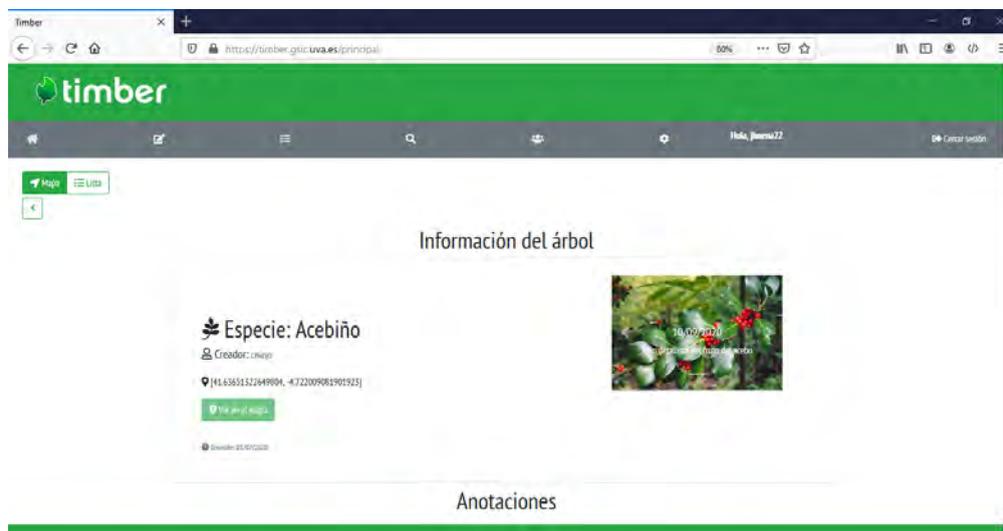
La separación de *frontend* y *backend* de la aplicación *Timber* ha permitido que este TFG se haya centrado en las características que el usuario puede ver de la interfaz y la forma de interactuar con la misma. Gracias a esta arquitectura multicapa, la interfaz de usuario está desacoplada de gran parte de la lógica del *backend* y permite poder integrar los datos expuestos en otras aplicaciones.

La elección de Angular como *framework* para la implementación del frontend, ha dado como resultado la creación de una aplicación web de una sola página (SPA) dinámica y de aspecto moderno. Además, el uso de *Bootstrap* ha agilizado mucho el proceso de diseño de la interfaz. La creación de servicios específicos para determinadas tareas ha facilitado la reutilización de código en varios componentes, por lo que se ha creado una gran estructura de servicios para abarcar todas las funcionalidades del *frontend*. También se hace uso de diferentes APIs de terceros para proporcionar otras funcionalidades, como la inserción del mapa interactivo con *Leaflet* en la aplicación.

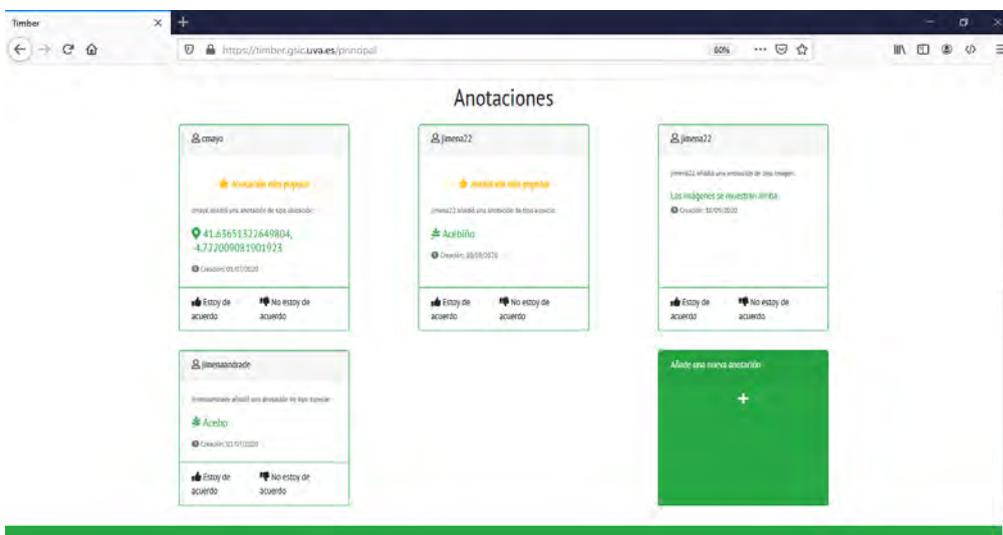
Lo que hace distinta a *Timber* con respecto a otras aplicaciones del estilo es la parte semántica. La implementación de la aplicación ha tenido que estar enfocada siempre a la traducción de los datos de entrada del usuario a la API de [May20]. El usuario introduce los datos como en cualquier otra aplicación, sin embargo, internamente y de manera transparente a él, la aplicación transforma esos datos a datos RDF según la ontología del apéndice A, permitiendo su almacenamiento en un almacén de triplas RDF. Hoy en día existen pocas aplicaciones web que realicen esta traducción, resultando un reto de programación para las participantes de este proyecto y una manera de dar mayor visibilidad a las tecnologías semánticas y su uso.

La parte social incluye modificaciones en los intercambios de datos con el *backend*, pues requieren de autenticación de los usuarios en cada comunicación con la API para poder crear/consultar datos almacenados. Esto añade complejidad principalmente en el *backend* ya que en el *frontend* tan solo supone añadir información del usuario en las cabeceras de las peticiones. Parte de la implementación de la parte social también consiste en controlar la entrada a determinados recursos expuestos (ver Tabla 4.1) que requieren el inicio de sesión para estar disponibles. Sin embargo, la parte social más relevante (curado de datos, votaciones de anotaciones, registro de actividad de usuarios) queda aún pendiente.

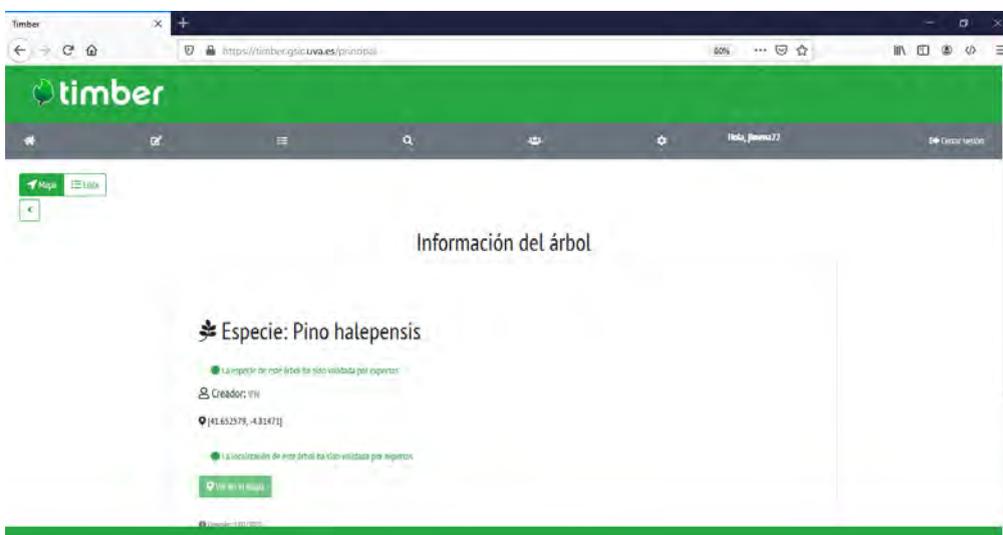
El resultado final es una aplicación web funcional, que permite añadir árboles en cualquier ubicación, facilitando además todas las especies de árboles que existen y dándote la posibilidad de subir alguna imagen de los mismos. La aplicación *Timber*, aunque en esta primera versión no posee toda la actividad social que se planteó en un primer momento, se puede decir que tiene carácter social, pues permite la creación de usuarios y que éstos puedan compartir la información de los árboles entre ellos. Además, la aplicación permite otras funcionalidades a usuarios registrados como completar con anotaciones árboles de otros usuarios, buscar árboles determinados mediante un buscador o un apartado de ajustes en el que ver su información personal e información relativa a *Timber* y sus creadores. La principal limitación existente en esta versión de la aplicación es el curado de los datos subidos, ya que por el momento no existe categorización de usuarios ni control de valoraciones de la información, provocando cierta incertidumbre en relación con la información expuesta. Otra limitación importante es la imposibilidad de editar o eliminar información almacenada en el *backend*. Todo ello está planteado como trabajo futuro en el capítulo siguiente.



(a) Datos primarios del árbol



(b) Anotaciones del árbol



(c) Ejemplo de árbol del IFN

Figura 4.8: Vista de la interfaz para el caso de uso *ConsultarArbolCompleto*

(a) Paso 1

(b) Paso 2

(c) Paso 3

(d) Paso 4

Figura 4.9: Vista de la interfaz para el caso de uso *RegistrarNuevoArbol*

Capítulo 5

Conclusiones y líneas de trabajo futuro

5.1. Conclusiones del trabajo realizado

Existen claras ventajas a la hora de utilizar aplicaciones sociales para fomentar el desarrollo y crecimiento de diferentes campos del conocimiento, además de que dichas aplicaciones utilizan los datos generados para facilitar nuestro día a día. Dentro del sector forestal, existen algunas iniciativas basadas en este carácter social, muchas de ellas encaminadas a almacenar datos de los diferentes árboles y plantas de nuestro alrededor. Sin embargo, ninguna de ellas facilita la anotación social de árboles. Además, el problema de este tipo de aplicaciones, al igual que la mayor parte de los sitios web, es la falta de herramientas para poder estructurar los datos que almacenan, evidenciando una falta de consistencia en los mismos y provocando la creación de “silos de datos” difícilmente reutilizables.

Con el objetivo de solventar estos problemas dentro de las aplicaciones del ámbito forestal, se decidió recurrir a las herramientas que proporciona la Web Semántica para reutilizar, en la medida de lo posible, sus vocabularios propios y, al comprobar que era necesario, se creó una ontología a mayores que completaba las necesidades del anotado social de árboles. Como resultado, se ha llegado a una solución social y semántica que explota y enriquece la Web de Datos, ya que hace uso de los Datos Abiertos Enlazados disponibles en la Web, a la vez que genera un conjunto de datos también abiertos y enlazados.

En este trabajo se aborda la parte de la aplicación propuesta que tiene que ver con el diseño y el desarrollo de la interfaz de usuario (*frontend*), con la característica principal de que los datos tratados en ella son transformados y adaptados como entradas a la API de [May20], donde son adaptados a lenguaje RDF para poder guardarlos en un almacén de triplas RDF. Se decidió utilizar el entorno de desarrollo de Angular para la creación de la aplicación web, ya que facilita en gran medida el diseño de aplicaciones responsivas y de una sola página.

La aplicación fue pensada para que permita a diversos tipos de usuarios buscar, geolocalizar, fotografiar y describir árboles de manera compartida. Las funcionalidades de la misma se pueden dividir en dos partes bien diferenciadas: (i) la parte dedicada a la recogida y exposición de información de los árboles y (ii) la parte social, dedicada a la

interacción entre los diferentes usuarios y el curado de los datos gracias a las opiniones de los mismos. Debido a las restricciones de tiempo para la elaboración de la misma, se decidió que la parte más importante de implementar era la primera, dejando a un lado el carácter social de la aplicación (pero no el colaborativo, pues son los usuarios los que suben las anotaciones a la aplicación). Sin embargo, aunque no se llegó a implementar, se ha hecho también el análisis y diseño de la parte social. El resultado final obtenido es una aplicación que permite a los usuarios crear árboles nuevos y completarlos con anotaciones de imágenes, especies y localización. La característica principal es que, de manera transparente al usuario, la aplicación estructura y transforma dichos datos según una ontología (Anexo A) diseñada con herramientas de la Web Semántica. Los datos de los usuarios registrados en la aplicación son transformados también y guardados en el almacén de triplas, siguiendo uno de los vocabularios predefinidos de la Web Semántica (*foaf*).

5.2. Líneas de trabajo futuro

Respecto al prototipo alcanzado en esta primera versión de la aplicación, se presentan varias líneas de trabajo por las que seguir. Por un lado, las mejoras de la parte de la interfaz ya implementada y, por otro, la parte social que no se ha llegado a implementar en este TFG pero de la que sí se realizó el análisis correspondiente.

Durante las pruebas realizadas para comprobar el rendimiento del prototipo de la aplicación alcanzado, se detectaron distintos problemas entre los que destacan la pérdida de rendimiento al navegar por el mapa durante un tiempo prolongado, provocando que la navegación por el mismo pierda fluidez. Esto es debido a que el mapa actualiza todos los datos de los árboles de la parte visible cada vez que se mueven las coordenadas, sean nuevos o no. Para solventarlo, se propone la creación de un algoritmo que detecte, en los momentos en los que se mueve el mapa, aquellas partes del mismo que son nuevas para actualizarlas. Es decir, cachear los datos para solo cargar los árboles de las zonas del mapa que no se veían en el instante anterior. De esta manera, las consultas a la API REST serán las mismas, pero el tráfico de datos disminuirá en gran medida, favoreciendo una mayor rapidez de carga y exposición de los elementos del mapa.

Por otro lado, se comprobó que los recursos expuestos por el *frontend* provocan que la aplicación sea algo opaca, debido a que se tiene una única URL para varias vistas (por ejemplo, para cualquier árbol). Esto dificulta la navegación a recursos específicos, como por ejemplo llegar a un árbol determinado sin tener que pasar por el mapa. Es por ello que se espera modificar las rutas de la aplicación de manera que se exponga una URL para cada recurso, centrándose sobre todo en el recurso Árbol.

Otro aspecto importante a tener en cuenta en futuras versiones es la implementación de las acciones de modificación y borrado de los datos, pues hasta el momento la aplicación solo permite su creación. Estas funcionalidades dependen, en gran medida, del desarrollo del *backend*. El borrado de los árboles deja a su vez un debate abierto y pendiente de solventar, ya que hay que plantearse si de deberían poder borrar árboles que tengan anotaciones asociadas.

Algunas otras mejoras (no tan relevantes) que se podrían hacer sobre el prototipo de esta versión son: (i) utilizar un *plugin* que permita al usuario subir las anotaciones de las

imágenes realizando la imagen en el momento de la creación o (ii) rediseñar el buscador de manera que se puedan filtrar los árboles mediante un mapa interactivo. El resto de tareas pendientes sobre esta versión de la aplicación están recogidas en el repositorio de *GitHub* del proyecto dentro del apartado de *Issues*.

Finalmente, se plantea la implementación del carácter social de la aplicación. El prototipo de esta versión permite crear usuarios, cuya función hasta ahora se limita a la publicación de árboles y anotaciones. Sin embargo, como se vio en el apartado de análisis, se pretende que dichos usuarios sean los encargados de validar la información de las diferentes anotaciones existentes. Además, se deja planteado un apartado de “Comunidad”, donde puedan interactuar entre ellos, ver las últimas anotaciones añadidas de personas de su interés o crear “Retos” en los que participar con otros usuarios.

Referencias

- [All11] Dean Allemang y Jim Hendler. *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier, 2ª edición, 2011.
- [Bat12] Jo Bates. This is what modern deregulation looks like: co-optation and contestation in the shaping of the uk's open government data initiative. *The Journal of Community Informatics*, 8(2):1–20, 2012.
- [Biz11] Christian Bizer, Tom Heath, y Tim Berners-Lee. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, chapter Linked Data: The Story So Far, páginas 205–228. IGI Global snippet, 2011.
- [BL01] Tim Berners-Lee, James Hendler, y Ora Lassila. The semantic web: Scientific american. *Scientific American*, 2001.
- [BL05] Tim Berners-Lee, Roy Fielding, y Larry Masinter. Rfc 3986: Uniform resource identifier (uri). RFC Editor, 2005.
- [BL06] Tim Berners-Lee. Linked data, 2006. [Website]. Último acceso: Junio 2009. Disponible en: <https://www.w3.org/DesignIssues/LinkedData.html>.
- [Boo07] Grady Booch, Robert Maksimchuk, Michael W. Engel, Bobbi J. Young, Jim Connallen, y Kelli A. Houston. *Object-oriented analysis and design with applications*, chapter 6, páginas 247–272. Addison-Wesley, 2007.
- [Bre09] John G Breslin, Alexandre Passant, y Stefan Decker. *The Social Semantic Web*. Springer Science & Business Media, 2009.
- [Cha11] Andre Charland y Brian Leroux. Mobile application development: web vs. native. *Communications of the ACM*, 54(5):49–53, 2011.
- [FG20] Belén Fierro García, Telmo Jurado, Guillermo Vega-Gorgojo, y José M. Giménez-García. Data exportation and publication - interim report. deliverable d2.2. Informe técnico, Cross-Forest Consortium, 2020.
- [Fie99] Roy Fielding, Jim Gettys, Jeffrey Mogul, Henrik Frystyk, Larry Masinter, Paul Leach, y Tim Berners-Lee. Rfc2616: Hypertext transfer protocol–http/1.1. RFC Editor, 1999.
- [Fie00] Roy T Fielding y Richard N Taylor. *Architectural styles and the design of network-based software architectures*, volumen 7. University of California, Irvine Irvine, 2000.
- [GG19] José Miguel Giménez García. Data collection, data quality, ontologies and vocabularies. deliverable d2.1. Informe técnico, Cross-Forest Consortium, 2019.

- [Hel12] Natalie Helbig, Anthony M Cresswell, G Brian Burke, y Luis Luna-Reyes. The dynamics of opening government data. *Center for Technology in Government*. [Online]. Disponible en: <http://www.ctg.albany.edu/publications/reports/opendata>, 2012.
- [IFN] Inventario forestal nacional. Gobierno de España, Ministerios de Agricultura, pesca y alimentación. [Website]. Último acceso 2020. Disponible en: <https://www.mapa.gob.es/es/desarrollo-rural/temas/politica-forestal/inventario-cartografia/inventario-forestal-nacional/>.
- [Jac00] Ivar Jacobson, Grady Booch, James Rumbaugh, y Salvadortr Sánchez. *El proceso unificado de desarrollo de software*. 2000.
- [Jad15] Madhuri A Jadhav, Balkrishna R Sawant, y Anushree Deshmukh. Single page application using angularjs. *International Journal of Computer Science and Information Technologies*, 6(3):2876–2879, 2015.
- [Joh14] Jeffrey Alan Johnson. From open data to information justice. *Ethics and Information Technology*, 16(4):263 – 274, 2014.
- [Kit14] Rob Kitchin. *The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences*. SAGE Publications, 2014.
- [Lan01] Doug Laney. 3d data management: Controlling data volume, velocity and variety. *META group research note*, 6(70):1, 2001.
- [Lok14] Rob Lokers, Stasinios Konstantopoulos, Armando Stellato, Rob Knapen, y Sander Janssen. Designing innovative linked open data and semantic technologies in agro-environmental modelling. *Disponible en: https://scholarsarchive.byu.edu/iemssconference/2014/Stream-A/45/*, 2014.
- [Lyt09] Miltiadis D Lytras y Patricia Ordonez de Pablos. *Social Web Evolution: Integrating Semantic Applications and Web 2.0 Technologies*. IGI global, 1st edición, 2009.
- [Mat04] Adam Mathes. Folksonomies: Cooperative classification and communication through shared metadata. *Graduate School of Library and Information Science, University of Illinois Urbana-Champaign*, 2004.
- [May20] Cristina Mayo. Diseño e implementación del servidor de una aplicación socio-semántica de anotación de árboles, trabajo fin de máster, máster en ingeniería de telecomunicación. *ETSI Telecomunicación. Universidad de Valladolid*, 2020.
- [Ope18] 2018. Open Knowledge Foundation - OPEN DATA HANDBOOK. [Website]. Último acceso: Mayo, 2018. Disponible en <http://opendatahandbook.org/>.
- [Pro13] Societize Project. Green paper on citizen science for europe: Towards a society of empowered citizens and enhanced research. Informe técnico, Societize consortium, 2013.
- [RC20] Adolfo Ruiz-Calleja, Miguel L Bote-Lorenzo, Guillermo Vega-Gorgojo, Sergio Serrano-Iglesias, Pablo García-Zarza, Juan I Asensio-Pérez, y Eduardo Gómez-Sánchez. Casuallearn: A smart application to learn history of art. En *European Conference on Technology Enhanced Learning*, páginas 472–476. Springer, 2020.

- [Res15] J Reschke. Rfc7617: The 'basic' http authentication scheme. RFC Editor, 2015.
- [Sny03] Carolyn Snyder. *Paper prototyping: The fast and easy way to design and refine user interfaces*. Morgan Kaufmann, 2003.
- [Ste15] Rod Stephens. *Beginning software engineering*. John Wiley & Sons, 2015.
- [VG20] Guillermo Vega-Gorgojo, José M. Giménez-García, Cristóbal Ordóñez, y Felipe Bravo. Pioneering easy-to-use forestry data with forest explorer. *Semantic Web Journal*, 2020. Under review.
- [Waz17] 2017. OpenDataSoft - Opendatasoft se asocia con Waze. [Website]. Último acceso: Julio , 2017. Disponible en <https://www.opendatasoft.com/es/blog/2017/07/24/opendatasoft-se-asocia-con-waze-para-entregar-una-iniciativa-de-ciudades-inteligentes-orientadas-al-transporte>.
- [Wil16] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1 – 9, 2016.
- [Wu06] Xian Wu, Lei Zhang, y Yong Yu. Exploring social annotations for the semantic web. En *Proceedings of the 15th international conference on World Wide Web*, páginas 417–426, Edinburgh Scotland, Mayo, 2006.
- [Yu07] Liyang Yu. *Introduction to the Semantic Web and Semantic Web Services*, chapter 1, páginas 3–36. Chapman and Hall/CRC, 2007.

Apéndice A

Ontología de la aplicación

En este apéndice se presenta la ontología del vocabulario utilizado para la aplicación de anotado socio-semántico diseñado en [May20].

```
1 @prefix sta: <http://timber.gsic.uva.es/sta/ontology/> . \#Sta => ...
   Social Tree App
2 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns/#> .
3 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema/#> .
4 @prefix xsd: <http://www.w3.org/2001/XMLSchema/#> .
5 @prefix owl: <http://www.w3.org/2002/07/owl/#> .
6 @prefix dc: <http://purl.org/dc/elements/1.1/> .
7 @prefix ifn: <http://crossforest.eu/ifn/ontology/> .
8 @prefix vann: <http://purl.org/vocab/vann/> .
9 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
10
11
12 # Se define la ontologia propia como rdf:type Ontology. Es decir, es ...
   miembro de la clase de owl Ontologia
13 sta: a owl:Ontology ;
14     rdfs:label "Social forest app Ontology"@en , "Ontologia de la ...
       aplicacion forestal social"@es ;
15     rdfs:comment "Ontology to publish data of the Social Forest ...
       inventory"@en , "Ontologia para publicar datos del Inventario ...
       Forestal Social" ;
16     owl:versionInfo 0.1 ;
17     dc:creator <https://www.gsic.uva.es/members/cmayo> ; #Etc...
18     dc:date "2019-10-12"^^xsd:date ;
19     owl:imports sta: , ifn: ; #Si hubiera que importar ontologias ...
20     vann:preferredNamespacePrefix "sta" .
21
22 #####
23 # Classes
24 #####
25
26 sta:Tree a owl:Class , rdfs:Class ;
27     rdfs:label "Tree"@en , "Arbol"@es ;
28     rdfs:comment "Woody perennial plant with an elongated stem ...
       supporting branches"@en , "Planta vivaz (que vive mas de dos ...
       anios, de tallo lenoso, que se ramifica a cierta altura del ...
       suelo"@es ;
29     rdfs:isDefinedBy sta: .
30
31 sta:TreePartPhoto a owl:Class , rdfs:Class ;
```

```

32  rdfs:label "Tree part photo"@en , "Parte del arbol de la foto"@es ;
33  rdfs:comment "Part of a tree that it is shown in the image"@en , ...
    "Parte del arbol que se muestra en la imagen"@es ;
34  rdfs:isDefinedBy sta: .
35
36  sta:Leaf a owl:Class , rdfs:Class ;
37  rdfs:subClassOf sta:TreePartPhoto ;
38  rdfs:label "Leaf"@en , "Hoja"@es ;
39  rdfs:comment "Leaf"@en , "Hoja de arbol"@es ;
40  rdfs:isDefinedBy sta: .
41
42  sta:Fruit a owl:Class , rdfs:Class ;
43  rdfs:subClassOf sta:TreePartPhoto ;
44  rdfs:label "Fruit"@en , "Fruto"@es ;
45  rdfs:comment "Fruit which grows on a tree"@en , "Fruto que crece en ...
    un arbol"@es ;
46  rdfs:isDefinedBy sta: .
47
48  sta:Trunk a owl:Class , rdfs:Class ;
49  rdfs:subClassOf sta:TreePartPhoto ;
50  rdfs:label "Trunk"@en , "Tronco"@es ;
51  rdfs:comment ""@en , ""@es ;
52  rdfs:isDefinedBy sta: .
53
54  sta:Branch a owl:Class , rdfs:Class ;
55  rdfs:subClassOf sta:TreePartPhoto ;
56  rdfs:label "Branch"@en , "Rama"@es ;
57  rdfs:comment ""@en , ""@es ;
58  rdfs:isDefinedBy sta: .
59
60  sta:Crown a owl:Class , rdfs:Class ;
61  rdfs:subClassOf sta:TreePartPhoto ;
62  rdfs:label "Crown"@en , "Copa"@es ;
63  rdfs:comment ""@en , ""@es ;
64  rdfs:isDefinedBy sta: .
65
66  sta:GeneralView a owl:Class , rdfs:Class ;
67  rdfs:subClassOf sta:TreePartPhoto ;
68  rdfs:label "General"@en , "General"@es ;
69  rdfs:comment "General view"@en , "Vista general"@es ;
70  rdfs:isDefinedBy sta: .
71
72  sta:Flower a owl:Class , rdfs:Class ;
73  rdfs:subClassOf sta:TreePartPhoto ;
74  rdfs:label "Flower"@en , "Flor"@es ;
75  rdfs:comment ""@en , ""@es ;
76  rdfs:isDefinedBy sta: .
77
78  sta:OtherPart a owl:Class , rdfs:Class ;
79  rdfs:subClassOf sta:TreePartPhoto ;
80  rdfs:label "Other plart"@en , "Otra parte"@es ;
81  rdfs:comment ""@en , ""@es ;
82  rdfs:isDefinedBy sta: .
83
84  sta:Image a owl:Class , rdfs:Class ;
85  rdfs:subClassOf foaf:Image ;
86  rdfs:label "Image"@en , "Imagen"@es ;
87  rdfs:comment "Tree's image"@en , "Imagen de un arbol"@es ;
88  rdfs:isDefinedBy sta: .
89

```

```

90     sta:Annotation a owl:Class , rdfs:Class ;
91     rdfs:label "Annotation"@en , "Anotacion"@es ;
92     rdfs:comment "Tree's annotation"@en , "Anotacion de un arbol"@es ;
93     rdfs:isDefinedBy sta: .
94
95     sta:SpeciesAnnotation a owl:Class , rdfs:Class ;
96     rdfs:subClassOf sta:Annotation ;
97     rdfs:label "Species annotation"@en , "Anotacion de especie"@es ;
98     rdfs:comment "Annotation about a tree species"@en , "Anotacion sobre ...
99     la especie de un arbol"@es ;
100    rdfs:isDefinedBy sta: .
101
102    sta:PositionAnnotation a owl:Class , rdfs:Class ;
103    rdfs:subClassOf sta:Annotation ;
104    rdfs:label "Position annotation"@en , "Anotacion de posicion"@es ;
105    rdfs:comment "Annotation about a tree position"@en , "Anotacion ...
106    sobre la posicion de un arbol"@es ;
107    rdfs:isDefinedBy sta: .
108
109    sta:ImageAnnotation a owl:Class , rdfs:Class ;
110    rdfs:subClassOf sta:Annotation ;
111    rdfs:label "Image annotation"@en , "Anotacion de imagen"@es ;
112    rdfs:comment "Annotation about a tree image"@en , "Anotacion sobre ...
113    la imagen de un arbol"@es ;
114    rdfs:isDefinedBy sta: .
115
116    #La anotacion primaria es la mas votada en un momento dado (los datos ...
117    que se consideran "validos")
118    sta:PrimaryAnnotation a owl:Class , rdfs:Class ;
119    rdfs:subClassOf sta:Annotation ;
120    rdfs:label "Primary annotation"@en , "Anotacion primaria"@es ;
121    rdfs:comment "Primary annotation about a tree"@en , "Anotacion ...
122    primaria sobre un arbol"@es ;
123    rdfs:isDefinedBy sta: .
124
125    sta:PrimaryPosition a owl:Class , rdfs:Class ;
126    rdfs:subClassOf sta:PrimaryAnnotation ;
127    rdfs:label "Primary position"@en , "Posicion primaria"@es ;
128    rdfs:comment "Primary position annotation about a tree"@en , ...
129    "Anotacion primaria de posicion sobre un arbol"@es ;
130    rdfs:isDefinedBy sta: .
131
132    sta:PrimarySpecies a owl:Class , rdfs:Class ;
133    rdfs:subClassOf sta:PrimaryAnnotation ;
134    rdfs:label "Primary species"@en , "Especie primaria"@es ;
135    rdfs:comment "Primary species annotation about a tree"@en , ...
136    "Anotacion primaria de especie sobre un arbol"@es ;
137    rdfs:isDefinedBy sta: .
138
139    sta:AssertedAnnotation a owl:Class , rdfs:Class ;
140    rdfs:subClassOf sta:PrimaryAnnotation ;
141    rdfs:label "Asserted annotation"@en , "Anotacion afirmada"@es ;
142    rdfs:comment "Annotation asserted by experts about a tree"@en , ...
143    "Anotacion afirmada por expertos sobre un arbol"@es ;
144    rdfs:isDefinedBy sta: .
145
146    sta:AssertedPosition a owl:Class , rdfs:Class ;
147    rdfs:subClassOf sta:AssertedAnnotation ;
148    rdfs:label "Asserted position"@en , "Posicion afirmada"@es ;
149    rdfs:comment "Annotation asserted by experts about a tree ...

```

```

    positiob"@en , "Anotacion afirmada por expertos sobre una ...
    posicion de un arbol"@es ;
142 rdfs:isDefinedBy sta: .
143
144 sta:AssertedSpecies a owl:Class , rdfs:Class ;
145 rdfs:subClassOf sta:AssertedAnnotation ;
146 rdfs:label "Asserted species"@en , "Especie afirmada"@es ;
147 rdfs:comment "Annotation asserted by experts about a tree ...
    species"@en , "Anotacion afirmada por expertos sobre una especie ...
    de un arbol"@es ;
148 rdfs:isDefinedBy sta: .
149
150
151 #####
152 # Object Properties
153 #####
154
155 sta:hasAnnotation a owl:ObjectProperty , rdf:Property ;
156 rdfs:label "has annotation"@en , "tiene anotacion"@es ;
157 rdfs:comment "annotation of a tree"@en , "anotacion de un arbol"@es ;
158 rdfs:domain sta:Tree ;
159 rdfs:range sta:Annotation ;
160 rdfs:isDefinedBy sta: .
161
162 sta:hasSpeciesAnnotation a owl:ObjectProperty , rdf:Property ;
163 rdfs:subPropertyOf sta:hasAnnotation ;
164 rdfs:label "has species annotation"@en , "tiene anotacion de ...
    especie"@es ;
165 rdfs:comment "annotation of species"@en , "anotacion de una ...
    especie"@es ;
166 rdfs:domain sta:Tree ;
167 rdfs:range sta:SpeciesAnnotation ;
168 rdfs:isDefinedBy sta: .
169
170 sta:hasPositionAnnotation a owl:ObjectProperty , rdf:Property ;
171 rdfs:subPropertyOf sta:hasAnnotation ;
172 rdfs:label "has position annotation"@en , "tiene anotacion de ...
    posicion"@es ;
173 rdfs:comment "annotation of a tree position"@en , "anotacion de una ...
    posicion de un arbol"@es ;
174 rdfs:domain sta:Tree ;
175 rdfs:range sta:PositionAnnotation ;
176 rdfs:isDefinedBy sta: .
177
178 sta:hasImageAnnotation a owl:ObjectProperty , rdf:Property ;
179 rdfs:subPropertyOf sta:hasAnnotation ;
180 rdfs:label "has image annotation"@en , "tiene anotacion de ...
    imagen"@es ;
181 rdfs:comment "annotation which includes a image"@en , "anotacion que ...
    incluye una imagen"@es ;
182 rdfs:domain sta:Tree ;
183 rdfs:range sta:ImageAnnotation ;
184 rdfs:isDefinedBy sta: .
185
186 sta:hasImage a owl:ObjectProperty , rdf:Property ;
187 rdfs:label "has image"@en , "tiene imagen"@es ;
188 rdfs:comment "image of a tree or a part of it"@en , "imagen de un ...
    arbol o de una parte"@es ;
189 rdfs:domain sta:ImageAnnotation ;
190 rdfs:range sta:Image ;

```

```

191   rdfs:isDefinedBy sta: .
192
193 #Propiedad mas general que especie (no es subpropiedad del ...
      ifn:hasTaxon porque tiene distinto dominio)
194 sta:hasTaxon a owl:ObjectProperty , rdf:Property ;
195   rdfs:label "has taxon"@en , "tiene taxon"@es ;
196   rdfs:comment ""@en , ""@es ;
197   rdfs:domain sta:SpeciesAnnotation ;
198   rdfs:range ifn:Taxon ;
199   rdfs:isDefinedBy sta: .
200
201 sta:hasPrimaryAnnotation a owl:ObjectProperty , rdf:Property ;
202   rdfs:subPropertyOf sta:hasAnnotation ;
203   rdfs:label "has primary annotation"@en , "tiene anotacion ...
      primaria"@es ;
204   rdfs:comment "primary annotation of a tree"@en , "anotacion primaria ...
      de un arbol"@es ;
205   rdfs:domain sta:Tree ;
206   rdfs:range sta:PrimaryAnnotation ;
207   rdfs:isDefinedBy sta: .
208
209 # Propiedad proxy, es la posicion que se considera buena para un arbol ...
      en cada momento
210 sta:hasPrimaryPosition a owl:ObjectProperty , rdf:Property ;
211   rdfs:subPropertyOf sta:hasPrimaryAnnotation ;
212   rdfs:label "has primary position annotation"@en , "tiene anotacion ...
      primaria de posicion"@es ;
213   rdfs:comment "primary annotation of a tree position"@en , "anotacion ...
      primaria de una posicion de un arbol"@es ;
214   rdfs:domain sta:Tree ;
215   rdfs:range sta:PrimaryPosition ;
216   rdfs:isDefinedBy sta: .
217
218 sta:hasPrimarySpecies a owl:ObjectProperty , rdf:Property ;
219   rdfs:subPropertyOf sta:hasPrimaryAnnotation ;
220   rdfs:label "has primary species annotation"@en , "tiene anotacion ...
      primaria de especie"@es ;
221   rdfs:comment "primary annotation of a tree species"@en , "anotacion ...
      primaria de una especie de un arbol"@es ;
222   rdfs:domain sta:Tree ;
223   rdfs:range sta:PrimarySpecies ;
224   rdfs:isDefinedBy sta: .
225
226 sta:hasAssertedAnnotation a owl:ObjectProperty , rdf:Property ;
227   rdfs:subPropertyOf sta:hasPrimaryAnnotation ;
228   rdfs:label "has asserted annotation"@en , "tiene anotacion ...
      afirmada"@es ;
229   rdfs:comment "annotation asserted by experts of a tree"@en , ...
      "anotacion afirmada por expertos de un arbol"@es ;
230   rdfs:domain sta:Tree ;
231   rdfs:range sta:AssertedAnnotation ;
232   rdfs:isDefinedBy sta: .
233
234 sta:hasAssertedPosition a owl:ObjectProperty , rdf:Property ;
235   rdfs:subPropertyOf sta:hasAssertedAnnotation ;
236   rdfs:label "has asserted position"@en , "tiene posicion afirmada"@es ;
237   rdfs:comment "annotation asserted by experts of a tree position"@en ...
      , "anotacion afirmada por expertos de una posicion un arbol"@es ;
238   rdfs:domain sta:Tree ;
239   rdfs:range sta:AssertedPosition ;

```

```
240   rdfs:isDefinedBy sta: .
241
242   sta:hasAssertedSpecies a owl:ObjectProperty , rdf:Property ;
243     rdfs:subPropertyOf sta:hasAssertedAnnotation ;
244     rdfs:label "has asserted species"@en , "tiene especie afirmada"@es ;
245     rdfs:comment "annotation asserted by experts of a tree species"@en , ...
        "anotacion afirmada por expertos de una especie de un arbol"@es ;
246     rdfs:domain sta:Tree ;
247     rdfs:range sta:AssertedSpecies ;
248     rdfs:isDefinedBy sta: .
249
250   # Para linkar el jpg con la imagen
251   sta:resource a owl:ObjectProperty , rdf:Property ;
252     rdfs:subPropertyOf foaf:depiction ;
253     rdfs:label "resource"@en , "recurso"@es ;
254     rdfs:comment ""@en , ""@es ;
255     rdfs:domain sta:Image ;
256     rdfs:isDefinedBy sta: .
```

Apéndice B

Glosario de la aplicación

Aficionado: nivel de usuario intermedio. Se accede a esta categoría mediante la obtención de puntos por logros. Los usuarios pertenecientes a este tipo pueden realizar todas las actividades de la aplicación, entre ellas: crear y consultar anotaciones, eliminar anotaciones propias, completar información de árboles existentes, localizar árboles por especies o usuario, etc.

Anotación: información relativa a un árbol que puede ser de tipo especie, localización o imagen.

Anotación primaria: anotación más votada por la comunidad de usuarios registrados. Se considera que es la información verdadera dentro de todas las anotaciones de ese tipo. Cuando se crea un árbol nuevo en el sistema, la información añadida pasa a ser la anotación primaria.

Anotar un árbol: acción que consiste en completar la información de un árbol existente dentro de la plataforma.

Experto: tipo de usuario correspondiente al máximo nivel dentro del sistema. Para pertenecer a este tipo, dichos usuarios deben haber alcanzado un umbral de puntos suficiente mediante logros. Los usuarios pertenecientes a este tipo pueden realizar las mismas acciones que los usuarios Principiante y Aficionado, pero sus validaciones en las anotaciones serán más valiosas.

Datos Abiertos Enlazados¹: método de publicación de datos que siguen una estructura particular para que puedan ser interconectados y reutilizados dentro de la Web.

Logro: acción realizada por un usuario consistente en crear un nuevo árbol, una nueva anotación o validar anotaciones de otros usuarios, entre otros. En función del tipo de logro realizado el usuario obtiene una determinada cantidad de puntos, que le sirven para subir de nivel.

Ontología: definición formal de tipos, propiedades, y relaciones entre entidades u objetos de un dominio en particular. La información de la aplicación se estructura según una ontología predefinida. Puede consultarse en el Apéndice A.

¹<https://www.w3.org/DesignIssues/LinkedData.html>

Principiante: nivel de usuario más bajo. Todos los usuarios pertenecen a este nivel cuando se dan de alta en el sistema. Los usuarios de este tipo pueden realizar todas las actividades de la aplicación.

SPARQL *Endpoint*: punto de presencia en la Web que es capaz de recibir y procesar las solicitudes del protocolo SPARQL. El punto SPARQL del sistema tiene la siguiente URL: <https://timber.gsic.uva.es/sparql>.

Usuario no registrado: usuario que utiliza la aplicación sin iniciar sesión en el sistema.

Usuario registrado: usuario que inicia sesión en la aplicación. Dentro de los usuarios registrados existen tres niveles: principiante, aficionado y experto.

Web Semántica: conjunto de actividades desarrolladas dentro de la Web cuyo objetivo es la creación de tecnologías para publicar datos legibles por máquinas.

Apéndice C

Fichero *Dockerfile*

```
1 # Stage 0, based on Node.js, to build and compile Angular
2 FROM node:latest as node
3 WORKDIR /app
4 COPY ./ /app/
5 RUN npm install
6 RUN npm run build -- --prod
7
8 # Stage 1, based on Nginx, to have only the compiled app,
9 # ready for production with Nginx
10 FROM nginx:alpine
11 COPY --from=node /app/dist/Jimena /usr/share/nginx/html
12 COPY ./nginx-custom.conf /etc/nginx/conf.d/default.conf
```


Apéndice D

Manual de usuario aplicación web *Timber*

D.1. Introducción

Timber es una aplicación web que le permitirá consular, crear y añadir árboles y sus características más relevantes de una manera colaborativa e interactiva con otros usuarios. Actualmente existen dos roles para acceder a la aplicación: **usuario registrado** y **usuario no registrado**.

D.2. Requerimientos

Los requerimientos mínimos para que la aplicación *Timber* funcione correctamente son los siguientes:

- a. Dispositivo (móvil, tableta u ordenador) con conexión a Internet.
- b. Navegador web Mozilla Firefox, Google Chrome o Safari.

El lenguaje predeterminado de la aplicación (y el único en esta primera versión) es el Español. La aplicación se adapta a cualquier tipo de dispositivo y tamaño de pantalla.

D.3. Instalación

La aplicación *Timber* no requiere de ninguna instalación adicional. Basta con acceder al enlace: <https://timber.gsic.uva.es/>¹.

D.4. Uso de la aplicación *Timber*

La vista de la página principal se muestra en la Figura D.1. En ella se pueden realizar tres acciones diferentes (disponibles en la barra de navegación gris del menú): navegar por el mapa para descubrir los árboles registrados en la aplicación, registrarse como nuevo usuario o iniciar sesión (si ya estás registrado).

Navegación por el mapa

¹Actualmente la URL no está disponible para los usuarios

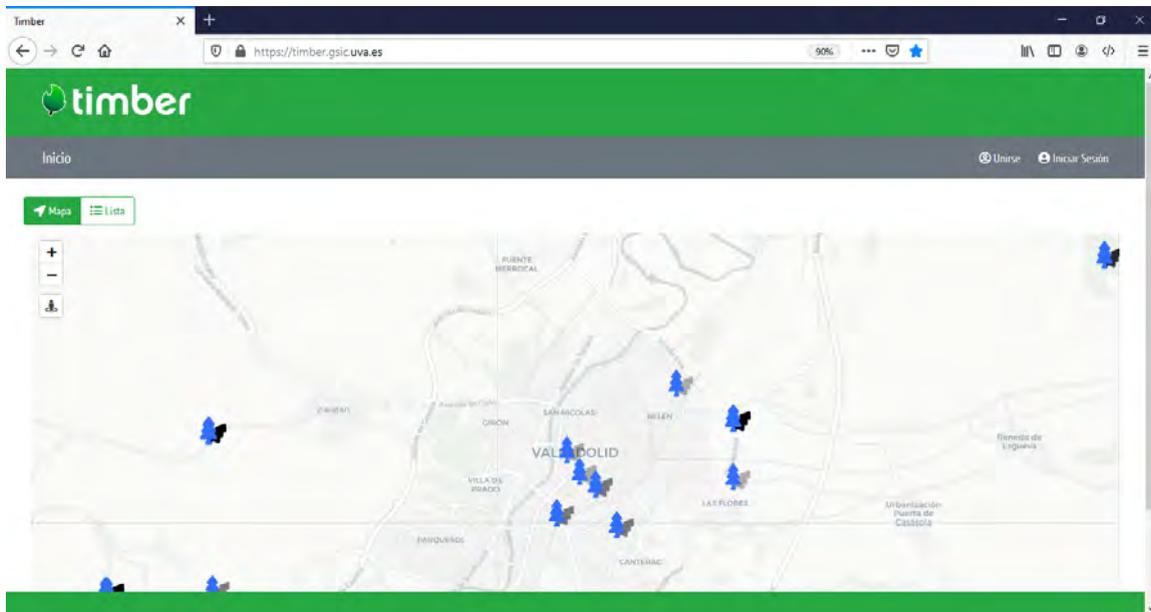


Figura D.1: Vista de la página principal de *Timber*

En la página principal se puede navegar por el mapa y ver los diferentes árboles según la zona en la que se encuentra, sin necesidad de estar registrado en la aplicación. Además, puede utilizar el icono de localización para que el mapa se sitúe en su ubicación actual (ver Figura D.2).

Si se hace zoom en una zona concreta, se puede pinchar en los diferentes árboles del mapa para ver el creador y la especie de los mismos, tal y como muestra la captura de la Figura D.3.

Si desea conocer la información completa de uno de los árboles, debe fijarse en el identificador del mismo y pinchar en el botón “Lista” (Figura D.4). Entonces, aparecerán en formato de lista todos los árboles que se encuentran en la zona del mapa actual. Debe encontrar el identificador del árbol escogido y pulsar “Ver más” (Figura D.5) para observar la información completa del árbol seleccionado.

La información puede incluir:

- Anotaciones primarias de localización y especie (marcadas con una estrella amarilla)
- Anotaciones validadas por expertos (marcadas con una estrella verde)
- Anotaciones de ubicación
- Anotaciones de especie
- Anotaciones de imagen

Las anotaciones son información acerca de la especie, la localización o imágenes de los árboles que otros usuarios han añadido. En todas estas anotaciones, adicionalmente se

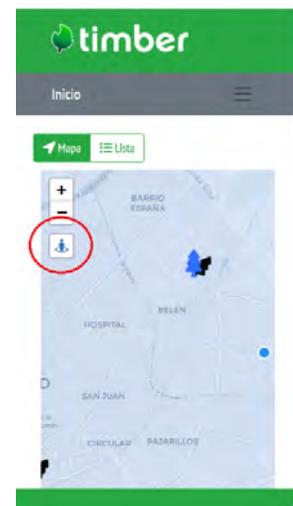


Figura D.2: Icono de geolocalización en el mapa

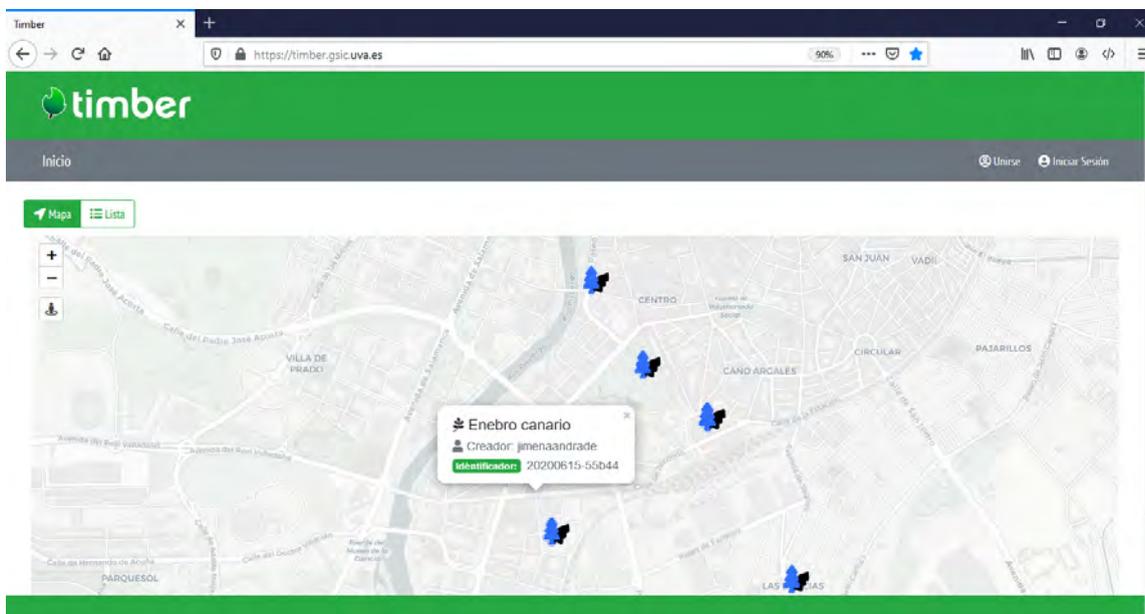


Figura D.3: *Popup* con información de creador y especie



Figura D.4: Botón para acceder a la lista de árboles

especifica el creador de las mismas, así como la fecha en que fueron creadas. Además, las anotaciones marcadas como “Anotación más popular”, corresponden a la anotación primaria del árbol (la más votada por los usuarios). Si se desea añadir una nueva anotación a cualquiera de los árboles existentes o crear un nuevo árbol necesitará registrarse en la aplicación.

Los botones de “Estoy de acuerdo” y “No estoy de acuerdo” no realizan ninguna acción en esta versión de la aplicación.

Registro

Si entra por primera vez en la aplicación y desea registrarse en el sistema, debe pinchar en el apartado “Unirse” de la barra gris del menú principal (Figura D.6). Entonces, le aparecerá un formulario que debe completar con sus datos personales. Una vez rellenados todos los campos, las barras rojas de la izquierda cambiarán de color a verde y se le permitirá pulsar el botón de “Enviar”.

El sistema comprobará que no existe ningún usuario con el mismo nombre de usuario o correo electrónico antes de realizar el registro. De ser así, serás registrado en el sistema y entonces serás redirigido a la página principal de usuarios registrados.

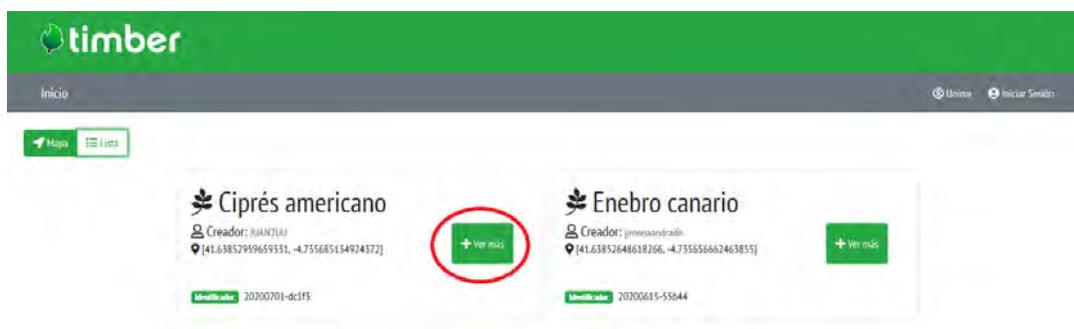


Figura D.5: Botón para acceder a la información de un árbol



Figura D.6: Vista parcial del menú de navegación

Inicio de sesión

Si ya tiene una cuenta de usuario en la aplicación, debe pinchar en el apartado “Iniciar Sesión” de la barra gris del menú principal (Figura D.6) e introducir su nombre de usuario y contraseña en el formulario.

FUNCIONALIDADES PARA USUARIOS REGISTRADOS

La página principal de usuarios registrados tiene un aspecto similar al de la página de inicio, pues se encuentra el mismo mapa y con las mismas funcionalidades. Sin embargo, la barra de navegación del menú es diferente y permite realizar diferentes funcionalidades. En la Figura D.7 se muestra la barra de navegación del menú. Las funcionalidades que ofrecen los diferentes iconos se explican a continuación:

1. **Inicio:** página principal de usuarios registrados. En ella se encuentra el mapa. Su aspecto es similar a la página de inicio, por lo que se pueden hacer las mismas funciones que las descritas en la sección de navegación por el mapa.
2. **Crear nuevo árbol:** esta sección te permitirá registrar un nuevo árbol en la aplicación. El único requisito para ello es especificar la localización del mismo.
3. **Mis anotaciones y mis árboles:** esta sección te permitirá ver los árboles que has creado y también las anotaciones de tus árboles y de otros árboles que hayas creado. En la parte de árboles, se permite además completarlos con nuevas anotaciones.
4. **Buscador:** permite buscar árboles que cumplan una determinada especie y/o un creador. Una vez buscados, se mostrarán en forma de lista para que puedas añadir más información si así lo deseas.
5. **Comunidad:** este apartado no se encuentra disponible en esta versión de la aplicación.
6. **Ajustes:** permite consultar tus datos personales. La opción de modificarlos no está disponible en esta versión. Además, muestra información relevante acerca de la aplicación, sus creadores y tu puntuación.

7. Cerrar sesión



Figura D.7: Menú de usuarios registrados