



Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques
Universitat de Barcelona**

La meva imatge és accessible?

Manuel Chamorro Borrallo

Director: Mireia Ribera

Realizado en: Departamento de
Matemática Aplicada y
Análisis

Barcelona, 20 de enero de 2016

Abstract

The project's documentation that will be developed in this final degree project involves a control system of the images' accessibility using a web application.

This project consist in the design and implementation of a web application where any kind of magazine could indicate the minim requirements that all image should fulfill to be allowed to publish that image, and where the author would upload his images to be analyzed and check if meet the requirements established by the publisher and if this image is accessible to people with visual problems. Also, authors will have access to a form where they can create an alternative text with the information showed in the image. This text will allow that people with visual problems have access to the same information thanks to an alternative path.

By developing this project we try to make an easy images' management for the publishers and the authors when they do a new article. At the same time, we try to make more accessible the content in this articles to people with any kind of visual problem.

Resumen

El proyecto documentado que se desarrollará en este Trabajo Fin de Grado consiste en un sistema de control de la accesibilidad de imágenes mediante una aplicación web.

El proyecto consiste en el diseño e implementación de una aplicación web donde cualquier tipo de revista o editorial pueda indicar cuales son los requisitos mínimos que deben cumplir las imágenes que vayan a ser publicadas en los artículos de las revistas, donde el autor pueda subir sus imágenes para ser analizadas y comprobar si cumplen los requisitos indicados por la editora y si dicha imagen es accesible para personas con problemas visuales. Además se les proporcionará a los autores un formulario para crear un texto alternativo con la información que se muestra en la imagen para que una persona con problemas para visualizarla pueda obtener la información que en ella se muestra por una vía alternativa.

Mediante el desarrollo de este proyecto se pretende facilitar la gestión de las imágenes a editoriales y autores a la hora de realizar la publicación de nuevos artículos y, a la vez, hacer más accesible el contenido de los nuevos artículos que se publiquen a personas con cualquier tipo de problema visual.

Resum

El projecte que es desenvoluparà en aquest Treball de Fi de Grau consisteix en un sistema de control d'accessibilitat d'imatges mitjançant una aplicació web.

El projecte consisteix en el disseny i implementació d'una aplicació web on qualsevol tipus de revista o editorial pugui indicar quins són els requeriments mínims que han de complir les imatges que vagin a ser publicades en els articles de les revistes, on els autors puguin pujar les seves imatges per ser analitzades i comprovar si compleixen els requeriments indicats per la editora i si la imatge és accessible per a persones amb problemes visuals. A més se'ls proporcionarà als autors un formulari per a crear un text alternatiu amb la informació que es mostra a la imatge per que una persona amb problemes per visualitzar-la correctament pugui obtenir la informació que en ella es mostra per una via alternativa.

Gràcies al desenvolupament de aquest projecte es pretén facilitar la gestió de les imatges a les editorials i als autors a l'hora de realitzar noves publicacions i, al mateix temps, fer més accessible el contingut de les publicacions a persones amb qualsevol tipus de problema visual.

Índice

1) Introducción	1
1.1 Legislación	1
1.2 Tipos de alteraciones visuales	2
1.3 Motivación	5
1.4 Objetivos	5
1.5 Estructura de la Memoria	6
2) Análisis de requisitos	7
2.1 Análisis de requisitos y funcionalidades	7
2.2 Requisitos funcionales	7
2.2.1 Casos de uso para el autor	8
2.2.2 Casos de uso para el editor	8
2.2 Requisitos no funcionales	9
2.3 Requisitos técnicos	10
3) Diseño	11
3.1 Arquitectura	11
3.2 Diseño de la base de datos	12
3.2.1 Modelo entidad-relación (ER)	13
3.2.2 Modelo de datos	14
3.2.3 Conexión	14
3.2.4 Consultas	15
4) Tecnologías utilizadas	16
4.1 Lenguajes	16
4.1.1 PHP	16
4.1.2 MySQL	16
4.1.3 HTML5, JavaScript y CSS3	17

4.1.3.1 HTML5	17
4.1.3.2 JavaScript	17
4.1.3.2.a JQuery	18
4.1.3.2.b AJAX	18
4.1.3.3 CSS3	18
4.2 Bootstrap	18
4.3 APIs utilizadas	20
4.3.1 ImageMagick	21
4.3.2 ABBYY Cloud OCR	22
4.3.3 Imagga	23
4.3.4 Cloudinary	24
4.4 Web Hosting	25
4.1.1 OpenShift	25
4.1.2 Otros web hosting	26
4.1.2.1 LAMP	26
4.1.2.2 Azure	26
4.1.3 ¿Por qué OpenShift?	27
4.5 Git	28
4.6 SublimeText	28
5) Planificación	29
6) Pruebas realizadas	31
6.1 Test de usuario	33
6.1.1 Resultados de los test	35
6.1.1.1 Autores	36
6.1.1.2 Editores	39
7) Conclusiones	41
7.1 Futuro	41

8) Bibliografía	44
9) Anexo	46
9.1 Anexo I: Contenido del ZIP	46
9.2 Anexo II: Manual de usuario	48
9.3 Anexo III: Casos de uso textuales	54
9.3 Anexo IV: Presentación del test de usuario	55
9.4 Anexo V: Consentimientos para los test de usabilidad	57

1. INTRODUCCIÓN

Todos los artículos que se realizan en las revistas, ya sean impresas o digitales, suelen utilizar imágenes para ilustrar su contenido o proporcionar ejemplos sobre el tema que se está tratando. Esto suele ser de gran ayuda para los lectores, ya que facilita la comprensión y la asimilación de la información que se está exponiendo en el artículo o proporciona una información indispensable para la comprensión de todo el artículo.

Cuando dichas imágenes no cumplen un mínimo de requisitos para poder ser entendibles o su formato supone un problema para los lectores con algún tipo de problema visual nos encontramos con diferentes dificultades. Que las imágenes no sean adecuadas supone un impedimento para que determinadas personas tengan un acceso completo a estos documentos y a la información que contienen. Por este motivo es necesario que tanto las editoras como los autores realicen un esfuerzo para que todo el contenido que publiquen sea accesible para todo el mundo.

Para poder facilitar la tarea de comprobar y convertir todas sus imágenes en información accesible, se ha decidido crear una aplicación web donde las editoriales puedan establecer cuales deben de ser las características mínimas que deben cumplir las imágenes que vayan a ser publicadas en la revista para que sean consideradas accesibles, y permitir a los autores comprobarlos con facilidad y ofrecerles herramientas para poder conseguir un mayor grado de accesibilidad.

1.1 Legislación

En la actualidad, en la legislación española existen diferentes normas relacionadas con el campo de la accesibilidad, buscan garantizar que todos los ciudadanos del estado pueden acceder con libertad e igualdad a todo tipo de información o documentación disponible. Esto incluye los artículos publicado en revistas.

La primera mención existente sobre la necesidad de eliminar los obstáculos y promover una sociedad igualitaria la encontramos en la Constitución Española de 1978, donde queda reflejado que *“Corresponde a los poderes públicos promover las condiciones para que la libertad y la igualdad del individuo y de los grupos en que se integra sean reales y efectivas; remover los obstáculos que impidan o dificulten su plenitud y facilitar la participación de*

todos los ciudadanos en la vida política, económica, cultural y social” [artículo 9.2]. Más concretamente, en el artículo 49 de la carta magna española se especifica que “Los poderes públicos realizarán una política de previsión, tratamiento, rehabilitación e integración de los disminuidos físicos, sensoriales y psíquicos, a los que prestarán la atención especializada que requieran y los ampararán especialmente para el disfrute de los derechos que este Título otorga a todos los ciudadanos” [artículo 49]. Estos dos artículos supusieron el primer paso en la legislación para la integración social de los discapacitados.

Más tarde en 2003 se redactó la LIONDAU o *ley 51/2003 de igualdad de oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad*, donde en su artículo número 10 se regula las condiciones mínimas de accesibilidad y no discriminación para alcanzar el mayor nivel de igualdad de oportunidades para todos los ciudadanos sin tener en cuenta sus discapacidades.

Esta ley fue derogada en 2013 por el Real Decreto Legislativo 1/2013 donde se aprueba el Texto Refundido de la Ley General de derechos de las personas con discapacidad y de su inclusión social. Este nuevo Real Decreto Legislativo unificó la LIONDAU con otras leyes de carácter similar, entre las que destaca la ley 13/1982, de 7 de abril, de integración social de los minusválidos.

1.2 Tipos de alteraciones visuales

A continuación expondremos de forma global algunas de las alteraciones visuales que existen y que están relacionadas con nuestro proyecto.

Pérdida de sensibilidad al contraste. La sensibilidad al contraste es la habilidad para distinguir entre variaciones de luz, su pérdida produce un efecto de desaparición de bordes, de objetos que se mezclan y la sensación de tener un halo grisáceo en el campo visual.



Figura 1. Ejemplo de pérdida de contraste.

Visión borrosa. Causa una dificultad para detectar o identificar objetos en un entorno con buenas condiciones visuales donde con una visión normal estos objetos podrían ser identificados perfectamente.



Figura 2. Ejemplo de visión borrosa.

Deslumbramiento. Es un fenómeno visual que causa una dificultad para diferenciar objetos debido a que produce un efecto de sobresaturación de la luz alrededor de los focos de luz o de las superficies que la reflejan.



Figura 3. Ejemplo de deslumbramiento.

Nieve visual. Este síntoma causa que la persona afectada vea como si estuviera nevando, especialmente con fondos oscuros. La densidad de la “nieve” varía dependiendo de la persona, en los casos más graves puede dificultar a la persona llevar una vida normal, incluso llegando a impedir que esta persona pueda leer.



Figura 4. Ejemplo de nieve visual.

Daltonismo. El daltonismo es un trastorno genético que provoca una dificultad para distinguir según qué tipo de colores. Al contrario de lo que cree mucha gente el daltonismo no solo implica no distinguir entre rojo y verde, sino que hay una gran variedad de grados de afectación que van desde la imposibilidad de distinguir entre cualquier tipo de color (acromatopsia) a una leve dificultad para distinguir matices de rojo, verde y azul. Esta afección supone un problema para los afectados en una gran variedad de ámbitos que van desde la vida cotidiana hasta la imposibilidad de realizar distintas profesiones como piloto o policía.



Figura 5. Imagen comparativa de diversos tipos de daltonismos por columnas. De izquierda a derecha: colores según una visión normal, según tritanopia, protanopia, deuteranopia o acromatopsia.

1.3 Motivación

La motivación que surge para la realización de este trabajo de fin de grado se puede dividir en dos apartados.

El principal sería la idea del proyecto en sí. Encuentro de vital importancia generar contenido que sea accesible a toda la población y es, por tanto, necesario crear nuevas ayudas tecnológicas que faciliten la clasificación de contenidos según si son accesibles o no para las personas con algún tipo de déficit (en este caso visual).

La creación de esta aplicación surge de la necesidad de las revistas y de los autores de tener un sistema sencillo y rápido para comprobar la accesibilidad de todas las imágenes que van a ser publicados en sus artículos y generar así textos alternativos si es necesario.

Por otra parte, la necesidad de crear esta aplicación me da la oportunidad de aplicar los conocimientos obtenidos durante mis estudios en el Grado de Ingeniería Informática en el completo desarrollo de una aplicación web creada a medida y específicamente para solventar las necesidades descritas previamente.

1.4 Objetivos

El objetivo principal de este proyecto es diseñar y desarrollar una pequeña aplicación web donde las editoriales y revistas puedan añadir qué requisitos mínimos de accesibilidad deben cumplir las imágenes que los autores quieran incluir en sus artículos y permitir a estos poder analizar sus imágenes de forma sencilla. Así podrán ver fácilmente qué requisitos cumplen, cuales deben modificar y se les ofrecerá un formulario para realizar un texto alternativo que incluya toda la información que se encuentra en la imagen para que los lectores que padecen algún tipo de deficiencia visual puedan acceder a ella por vía de este texto.

Los objetivos principales que se esperan lograr al finalizar este proyecto son los siguientes:

- Desarrollar un interfaz fácil de entender y utilizar.
- Proporcionar un formulario intuitivo para poder añadir nuevas revistas para la comprobación de requisitos.
- Extraer toda la información necesaria de la imagen para el posterior análisis.

- Realizar un OCR para detectar texto en la imagen.
- Facilitar que el usuario pueda comprobar el contraste entre colores.
- Modificar las imágenes para simular como serian visualizadas por una persona con daltonismo.
- Proporcionar un formulario para generar textos alternativos.

1.5 Estructura de la Memoria

Una vez introducido el tema de este trabajo pasamos a detallar como se desarrollará este proyecto.

La estructura de la memoria será la siguiente:

- Análisis de requerimientos: Analizaremos los requisitos necesarios para el desarrollo de nuestra web y las funcionalidades a las que tendrá acceso cada uno de nuestros tipos de usuario.
- Diseño: Durante este capítulo nos centraremos en explicar las características que han sido desarrolladas para la aplicación, como puede ser el diseño o los análisis que se realizan a las imágenes.
- Tecnologías utilizadas: Este apartado va a estar centrado en explicar todas las tecnologías que han sido utilizada para el desarrollo y para que funcionalidades de nuestros objetivos han sido necesarias.
- Planificación: Durante este capítulo se explicara cómo ha sido planificado y distribuido el desarrollo del trabajo durante el tiempo de implementación.
- Pruebas realizadas: Aquí se explicara todas las pruebas que se han realizado para comprobar el correcto funcionamiento de nuestra aplicación.
- Conclusiones: Explicaremos las conclusiones que hemos sacado de nuestro trabajo y cuales deberían ser los pasos a seguir para el futuro de la aplicación.
- Bibliografía.
- Anexo.

2. ANÁLISIS DE REQUISITOS

Antes de comenzar el desarrollo de la página web que albergará nuestro proyecto se han analizado los requisitos y funcionalidades que este proyecto debía cumplir.

A continuación explicaremos más detenidamente este análisis.

2.1 Análisis de requisitos y funcionalidades

El requisito principal de este proyecto es el desarrollo de un prototipo de un sistema de control para la accesibilidad de imágenes mediante una aplicación web.

De esta forma se facilitará, tanto a editores como a autores, la gestión de la accesibilidad de sus imágenes. Para ello se les proporcionará a ambos, editores y autores, las herramientas necesarias para lograr este objetivo.

Las imágenes que serán publicadas en las revistas deberán cumplir una serie de requisitos mínimos en algunas de sus características, esto son:

- Formato: tener uno de los formatos (.jpg, .png...) que acepta la revista.
- Tipo: Cumplir uno de los tipos de imagen (color, blanco y negro,...) aceptados.
- Resolución: superar la resolución mínima.
- Tamaño: Tener un tamaño mayor que el mínimo.
- Peso: No superar el peso máximo.

2.2 Requisitos funcionales

Esta aplicación, como ya se ha comentado con anterioridad, está dirigida a dos grupos de usuarios distintos.

- Autores: Este usuario es el que utilizará la aplicación para comprobar la accesibilidad de su imagen.
- Editores: Son los responsables de las revistas, establecen los requisitos que la herramienta usará como filtros para validar las imágenes.

En los puntos siguientes se detallarán los casos de uso para cada uno de nuestros usuarios. En el anexo III se pueden consultar con más detalle los casos de uso textuales para la aplicación.

2.2.1 Casos de uso para el autor

Este usuario es el que utilizara la aplicación para comprobar la accesibilidad de su imagen y tendrá acceso a un formulario para generar un texto alternativo que muestre la información necesaria que hay en dicha imagen.

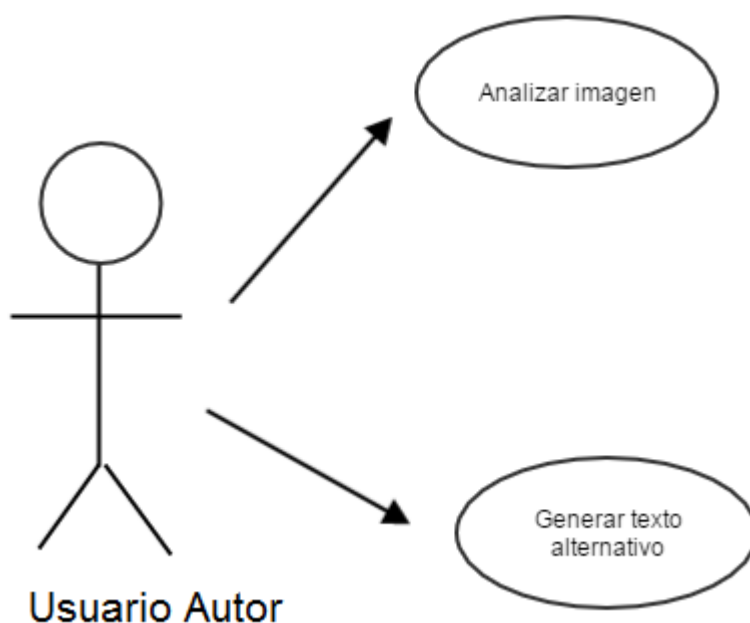


Figura 6. Casos de uso para el autor.

2.2.2 Casos de uso para el editor

Son los responsables de las revistas. Se encargarán de subir a nuestra base de datos, mediante un formulario, la características necesaria que deberán cumplir las imagen que los autores vayan a publicar en sus revistas. De forma puntual podrán utilizar las herramientas para los autores para validar imágenes.

Estas características serán las que se tengan en cuenta para considerar las imágenes accesibles o no para dicha revista.

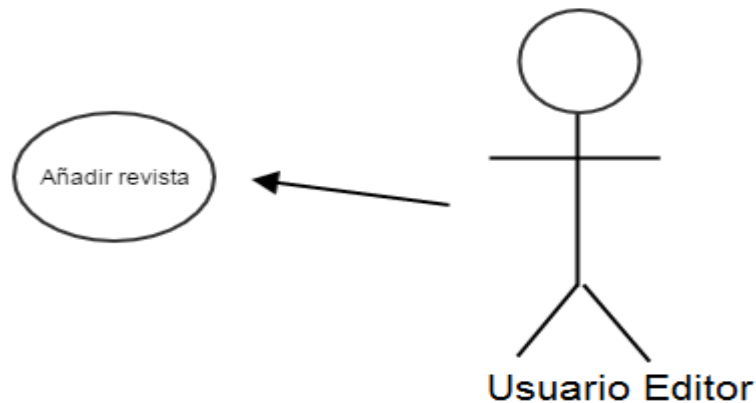


Figura 7. Casos de uso para el editor.

2.2 Requisitos no funcionales

Nuestra aplicación utilizara un diseño web responsivo para poder ser accesible a través de ordenadores, teléfonos móviles o tablets. Este tipo de diseño permite que la web se pueda adaptar a todos estos dispositivos sin tener en cuenta el tamaño de su pantalla y sin tener que crear una aplicación específica para ellos.

La interfaz deberá ser sencilla, poco recargada y creada con colores agradables a la vista. El tamaño de letra usado debe tener un tamaño adecuado para facilitar su lectura. También tendrá que ser lo más intuitiva posible para que los usuarios puedan comprender con facilidad el funcionamiento de la web y, de esta forma, conseguir evitar que su interés por nuestra página pueda decaer por no entender bien su funcionamiento.

Dado a que el tratamiento de imágenes es un proceso muy costoso, sobre todo para imágenes de gran tamaño, se ha buscado hacer este proceso lo más eficiente posible para no tener al usuario en espera durante largos periodos mientras se realiza el análisis (Ya que estos tiempos de espera pueden causar la perdida de interés en el usuario sobre el beneficio de utilizar nuestra aplicación).

2.3 Requisitos técnicos

Para que toda la aplicación pueda funcionar correctamente ésta necesita cumplir con unos requisitos técnicos.

En el lado del back-end vamos a necesitar un servidor para alojar nuestra aplicación. Esto puede solucionarse contratando un hosting en internet. Entre la gran variedad de ofertas que se pueden encontrar, nuestra elección será explicada posteriormente.

Además, también será necesario contar con una base de datos conectada a nuestro proyecto para almacenar toda la información que se considere necesaria.

Para la estructuración de la aplicación web nos basaremos en la estructura de tres capas que utilizan el resto de páginas web.

3. DISEÑO

En este apartado realizaremos la explicación de cómo está estructurada nuestra página web y comentaremos la arquitectura del proyecto. Es decir, explicaremos el diseño de la base de datos y cómo accedemos a los datos almacenados en ella. También se explicará la estructura en la que se almacenan todos los ficheros necesarios para el funcionamiento de la aplicación.

Anteriormente se ha comentado que nuestra aplicación web está dividida en dos partes fundamentales para su funcionamiento, el front-end, que será la parte visual que vea el usuario en su navegador, y el back-end que estará localizado en el servidor.

El front-end, estará localizado en el lado del cliente. Estará diseñado con CSS y HTML5, también utilizaremos JavaScript (especialmente librería JQuery) y AJAX para realizar una interacción con el usuario lo más fluida posible.

La segunda parte está compuesta por el back-end, está localizado en el lado del servidor. Aquí será donde se ejecuten los procesos más costosos de la aplicación, más concretamente los análisis de las imágenes. Para su desarrollo utilizaremos PHP, explicado más detalladamente en próximos apartados. En el back-end también será el lugar donde se realicen los accesos a la base de datos que vamos a utilizar, y que se basará en MySQL.

3.1 Arquitectura

La arquitectura de nuestro proyecto está dividida en tres partes, estas partes corresponden al cliente, el servidor (junto a la base de datos) y, por último, las APIs (a las que accederemos para realizar diferentes partes del análisis de las imágenes). Éstas últimas se explican en profundidad en el apartado *4. Tecnologías utilizadas*.

Los usuarios tendrán acceso a la interfaz de la web, desde la cual podrán realizar peticiones HTTP al servidor. El servidor, teniendo en cuenta la petición, realizará las acciones acordadas que se hayan programado para esta petición en concreto. Estas acciones pueden ser, desde ejecutar un código en PHP, realizar un acceso a la base de datos para añadir, modificar o consultar información o realizar una petición para utilizar una de las APIs externas para lo que utilizará una nueva consulta HTTP.

La petición realizada por el cliente también puede conllevar una combinación de las distintas acciones que se acaban de mencionar.

Una vez que el servidor termine de realizar la cantidad de procesos necesarios enviará al cliente una respuesta HTTP con la información conveniente para esa petición.

En la siguiente imagen se puede ver una representación visual que muestra la arquitectura de la aplicación web tal y como se acaba de explicar.

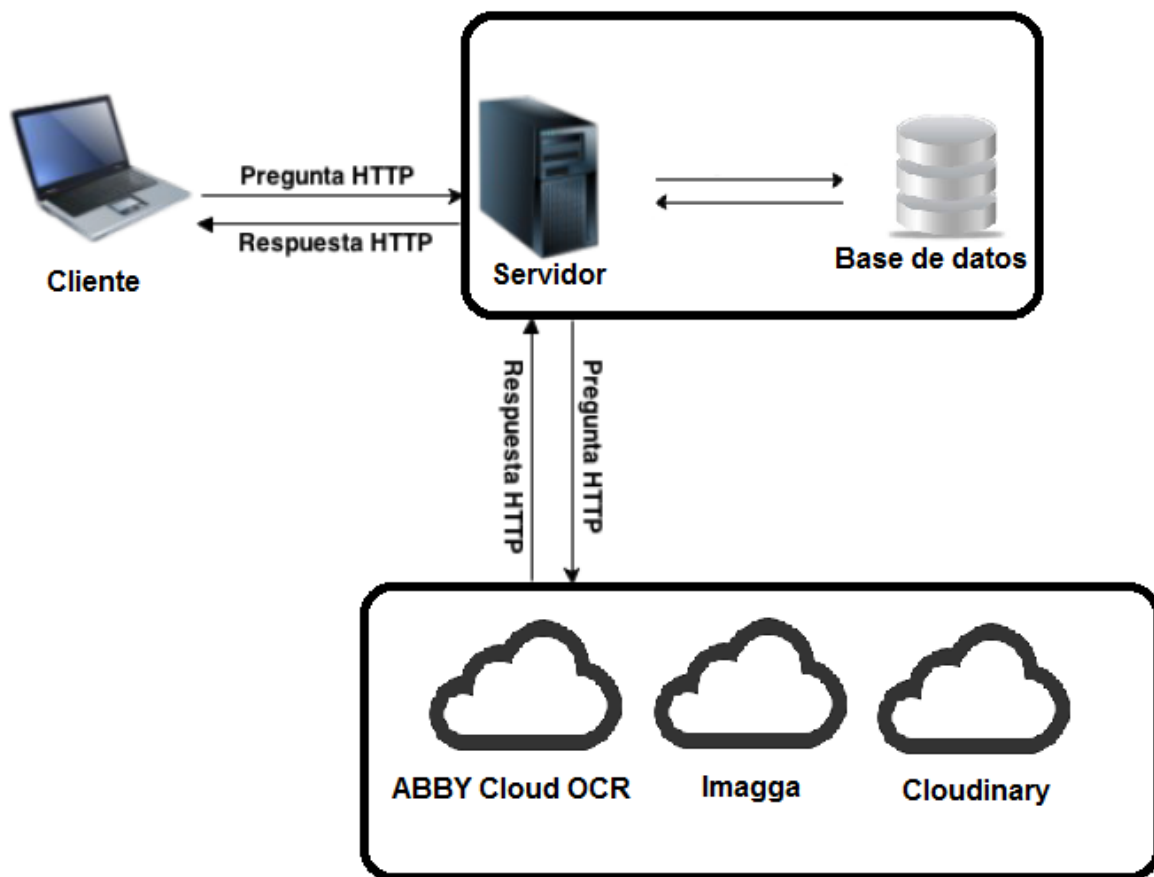


Figura 8. Arquitectura de la web.

3.2 Diseño de la base de datos

La base de datos será el lugar de nuestro servidor donde se almacenará toda la información para cada revista que se añada a nuestra aplicación mediante el formulario que se proporciona a los editores.

3.2.1 Modelo entidad-relación (ER)

En la siguiente imagen se ilustra el diagrama de entidad-relación de la base de datos.

En este diagrama se puede observar de qué tablas y atributos consta nuestro proyecto y cómo están relacionados entre sí.

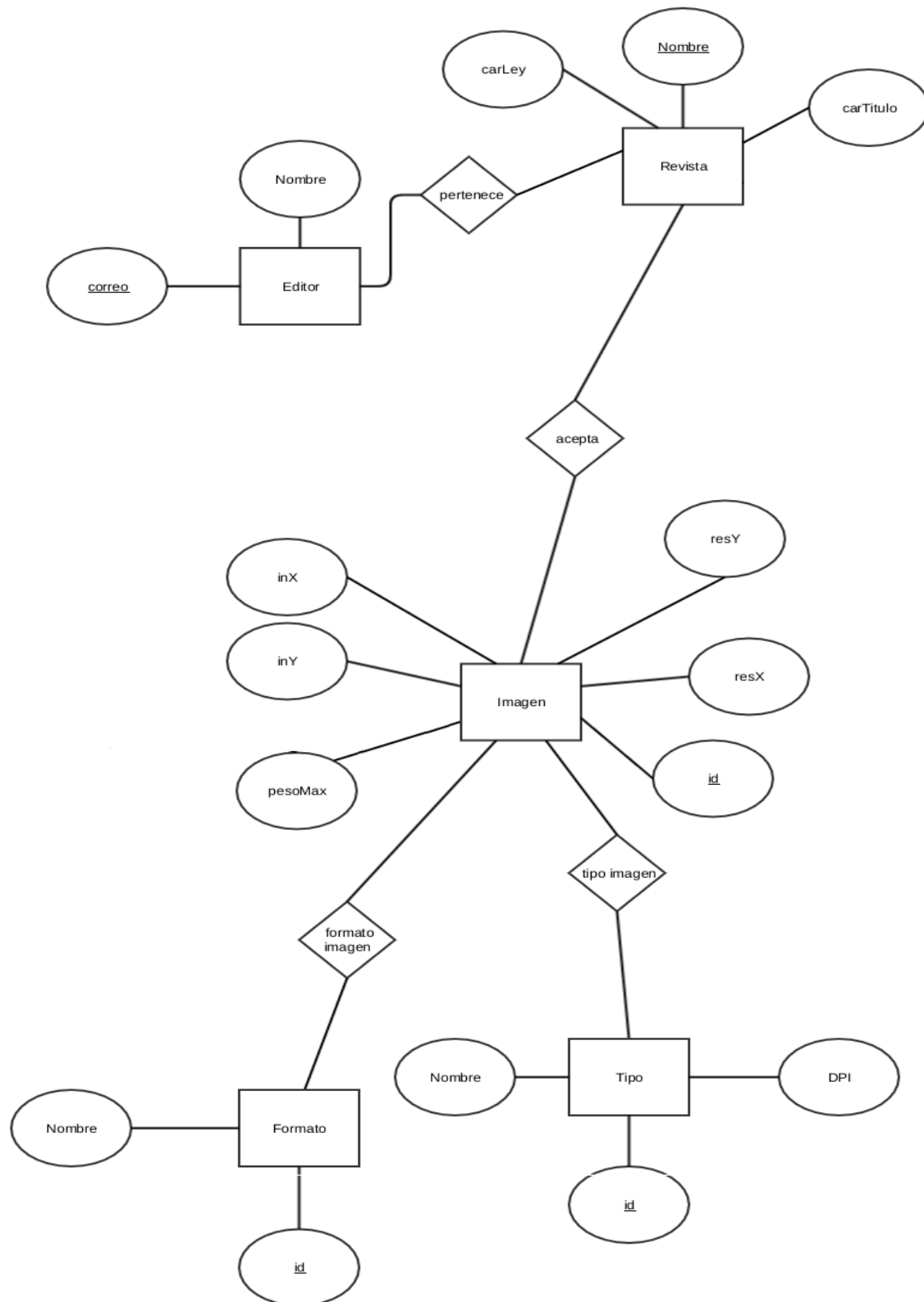


Figura 9. Modelo entidad-relación de la base de datos.

3.2.2 Modelo de datos

En la imagen 10 se puede observar el modelo relacional de la base de datos resultante del modelo entidad-relación de la base de datos (mostrado anteriormente). En este modelo relacional se puede observar las distintas tablas que forman la base de datos, los atributos que componen cada una de ellas y como se relacionan entre sí.

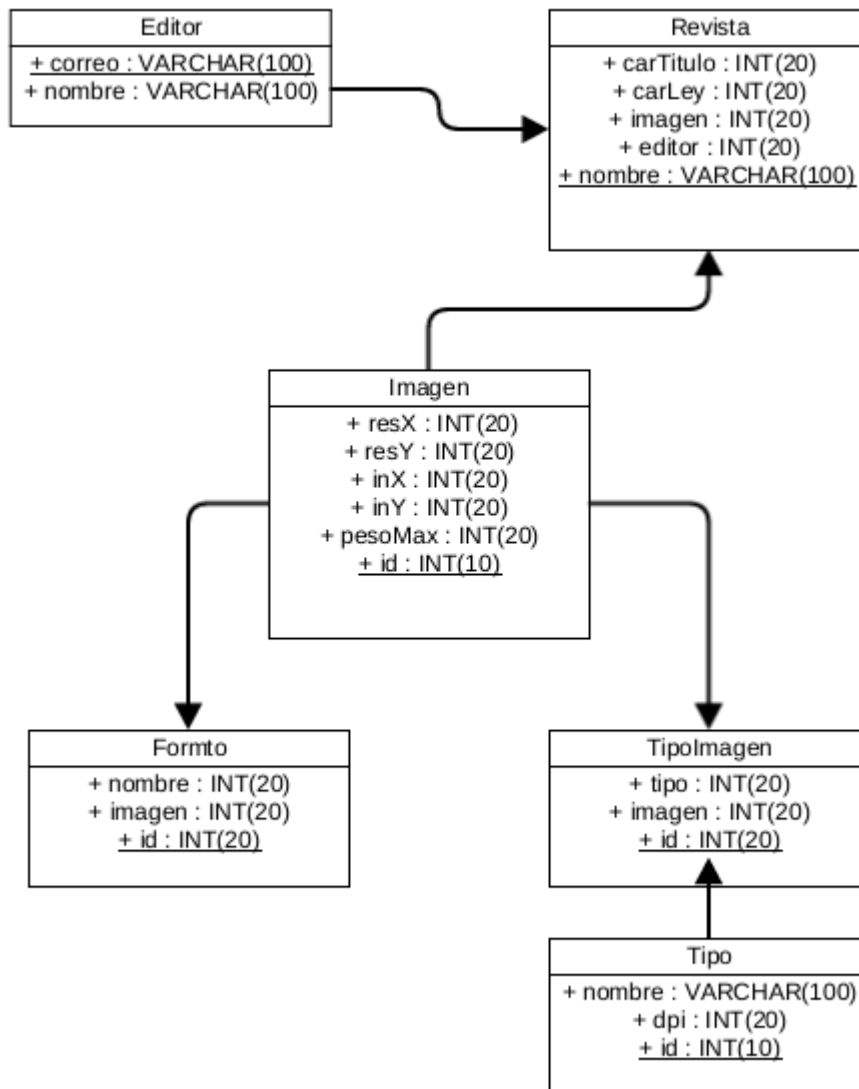


Figura 10. Modelo relacional de la base de datos.

3.2.3 Conexión

Para realizar la conexión entre la base de datos y la aplicación son necesarios una serie de parámetros proporcionados por el host de nuestra aplicación. En nuestro caso este host será OpenShift (del que hablaremos más adelante en el apartado 4.1.1).

Los parámetros necesarios para realizar la conexión serán los siguientes:

- **Dirección IP:** La dirección IP, junto al puerto, donde está localizada nuestra base de datos.
- **Nombre:** El nombre de la base de datos.
- **Usuario:** El nombre de usuario de la base de datos.
- **Contraseña:** La contraseña de la base de datos.

Con toda esta información ya es posible realizar una conexión a la base de datos mediante la consulta correspondiente y acceder a la información.

Por motivos de seguridad y de fluidez la conexión con la base de datos solo deberá estar abierta en el momento en que se vaya a realizar una consulta.

3.2.4 Consultas

Para poder consultar los datos almacenados en la base de datos es necesario utilizar la conexión que se establece con la configuración que se ha explicado en el punto anterior. Estas consultas se realizarán desde los ficheros PHP mediante *queries* de SQL. Después éstas nos devolverán una lista con los datos solicitados.

El tipo de consultas que se pueden realizar a la base de datos son cuatro:

- **INSERT:** Nos permite añadir nueva información en una tabla.
- **SELECT:** Permite leer la información almacenada previamente.
- **UPDATE:** Modifica la información que ya existe en una tabla.
- **DELETE:** Sirve para eliminar datos de la base de datos.

4 TECNOLOGÍAS UTILIZADAS

Una vez realizado, en los apartados anteriores, el análisis de todos los requerimientos nos disponemos a explicar los lenguajes y las herramientas que han sido necesarias para desarrollar el proyecto y conseguir alcanzar los objetivos deseados.

4.1 Lenguajes

Primero hablaremos de los lenguajes de programación que han sido utilizados para el desarrollo del proyecto.

Se explicarán tanto los lenguajes utilizados para el back-end (lado del servidor) como para el front-end (lado del cliente).

4.1.1 PHP

El PHP o Hypertext Preprocessor es un lenguaje de programación de uso general diseñado originalmente para el desarrollo de contenido dinámico para la web y que puede ser ejecutado en el lado del servidor.

PHP tiene la característica de soportar un gran número de tipos de base de datos distintas, incluyendo MySQL que será la que utilizaremos en nuestro caso.

La razón de haber usado este lenguaje es que se trata de uno de los lenguajes más flexibles y de alto rendimiento, en constante evolución, con un gran número de extensiones del lenguaje y que nos ha sido fácil de aprender debido a su gran número de similitudes con Java.

4.1.2 MySQL

MySQL es un gestor de datos de tipo relacional. Se trata de un gestor de bases de datos de programación, operacional, multihilo y multiusuario. Una base de datos es una colección estructurada de tablas que contiene información. Para acceder a dichas tablas y consultar o modificar la información en ellas se utiliza este gestor.

Se caracteriza por su rapidez, su fácil adaptación a cualquier tipo de necesidad y requerimiento y por permitir su uso a través de la web o de diferentes lenguajes de programación.

Este es el lenguaje que utilizaremos para gestionar nuestra base de datos combinado con PHP, lenguaje que, como ya se ha comentado, utilizaremos en el lado del servidor. Nos permite guardar y gestionar toda la información guardada necesaria para el buen funcionamiento de este proyecto.

4.1.3 HTML5, JavaScript y CSS3

Para la realización del front-end (parte del cliente) utilizaremos la combinación de HTML5, para estructurar y presentar el contenido en la web, JavaScript para definir la interacción con el usuario y CSS3 para el diseño.

4.1.3.1 HTML5:

Se trata de la última versión de HTML.

HTML es un lenguaje de marcas diseñado para estructurar y presentar contenido en una página web. La principal mejora de HTML5 respecto a versiones anteriores es que ha sido creado para implementar funcionalidades nativamente sin necesidad de plugins, entre las que más destacan se encuentran:

- Geolocalización: Nos permite obtener la posición donde está localizado el usuario
- Drag & Drop: Proporciona un API para implementar fácilmente en páginas web la funcionalidad de “arrastrar y soltar”.
- Local Storage: Permite guardar información en una carpeta usada por el navegador en el disco duro del usuario sin las limitaciones de las cookies (falta de privacidad, empeoramiento de rendimiento por su inclusión en cada petición al servidor y almacenamiento muy limitado).
- Reproducción de vídeo y audio.

4.1.3.2 JavaScript

Es un lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente como implementación para el lado del cliente como parte de un navegador web, permitiendo webs más dinámicas y mejoras en la interfaz de usuario. El navegador es el encargado de interpretar y ejecutar el código JavaScript.

4.1.3.2.a JQuery

Se trata de una de las librerías para JavaScript más conocidas, y gracias a su facilidad de uso y su gran potencial se ha convertido en un complemento casi indispensable en el desarrollo web. Además hay que tener en cuenta que JQuery es software libre y de código abierto.

La característica principal de la librería es permitir cambiar el contenido de una página web sin necesidad de recargarla, gracias a la manipulación del DOM y a la utilización de AJAX.

4.1.3.2.b AJAX

Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente mientras se mantiene la comunicación asincrónica con el servidor, de esta forma se pueden realizar cambios en la página web sin necesidad de recargas, mejorando la usabilidad, velocidad e interacción con el usuario.

4.1.3.3 CSS3:

Se trata de un lenguaje utilizado para crear y definir la presentación de un documento estructurado escrito en HTML o XML, en nuestro caso lo utilizaremos para la presentación de las distintas páginas que formaran nuestra aplicación web.

La utilidad detrás del desarrollo de CSS es la de poder separar la estructura de un documento de su diseño, facilitando el desarrollo de ambos y de sus posibles modificaciones.

4.2 Bootstrap

Bootstrap es un framework de desarrollo, libre y de código abierto diseñado para crear aplicaciones web.

Se trata del framework más popular para HTML y CSS por contener templates para formularios, botones, navegación y otros muchos componentes de la interfaz. Además, opcionalmente, se le pueden añadir extensiones para JavaScript.

Bootstrap ha sido diseñado para facilitar el desarrollo de páginas web dinámicas y crear proyectos web que cumplan con un diseño responsivo y con la filosofía mobile-first (ver figura 11).



Figura 11 Representación del concepto mobile-first.

El concepto mobile-first consiste en asignar prioridades a la información para ser ordenada según la importancia que tenga sobre el resto de información. Esto es debido a que la información será mostrada en una pantalla de reducido tamaño, ya que primeramente nos centramos en dispositivos móviles. Esta técnica nos ayuda a crear un diseño más eficaz, organizado, sencillo y responsivo

Creado y desarrollado por el equipo de Twitter para poder acelerar sus proyectos y poder compartir patrones de diseño entre los miembros dentro de su propia compañía.

A mediados de 2011 fue liberado como proyecto de código abierto y rápidamente se convirtió en uno de los proyectos más populares de Github, llegando a ocupar el primer puesto en Febrero de 2012.

Su gran éxito en la actualidad se debe a que cualquier web ya no es solo consultable desde un ordenador, sino que además se puede llegar a ella desde una gran variedad de dispositivos móviles, como smartphones o tablets, con una gran variedad de tamaños y resoluciones haciendo imposible el desarrollo individual para cada uno de ellos. A causa de la aparición de esta gran cantidad de dispositivos distintos, nació en la comunidad de desarrolladores la necesidad de encontrar una solución para que las webs pudiesen adaptarse a cualquier tipo de dispositivo sin necesidad de un diseño específico para cada

uno. Fue entonces cuando aparecieron las tendencias del diseño web responsivo y la filosofía mobile-first.

Nuestro diseño para la aplicación web se ha basado en el framework de Bootstrap para poder adaptar su diseño a todo tipo de dispositivos con un navegador web. Esto ha facilitado en gran medida la creación de las vistas de nuestra página ya que este framework se encarga automáticamente de realizar las adaptaciones a tamaño de pantalla en el que se esté ejecutando, permitiendo tener solo una hoja de estilos en común para todos los dispositivos (en vez de tener que crear una propia para cualquiera de los tipos de dispositivos que se puedan encontrar en el mercado).



Figura 12 Diseño de web responsivo.

4.3 APIs utilizadas

En este apartado vamos a hablar de las distintas APIs que han sido necesarias utilizar para lograr el objetivo principal de nuestro proyecto.

Estas APIs han sido todas diseñadas para la manipulación y análisis de imágenes aunque cada una de ellas tiene una función específica, que se detallará a continuación, como puede ser la realización de un OCR (Optical Character Recognition) o la modificación de los colores de la imagen para simular como sería vista por una persona con daltonismo.

Para la selección de cada una de estas APIs se ha buscado y comparado entre diversas APIs con similares características y se ha elegido la que podía ser mejor para nuestro

proyecto. Los criterios que hemos utilizado para seleccionar cada API entre sus similares han sido:

- Gratuito: Nos hemos centrado en buscar APIs que fuesen gratuitas o que ofreciesen planes gratuitos para desarrolladores.
- Prestaciones: Entre las que cumplían el criterio anterior, hemos elegido la que nos proporcionaba todas las prestaciones que necesitábamos y si ofrecía alguna más que nos pudiese acabar siendo útil.
- Mayor calidad: Por último, elegimos las APIs que proporcionaban mejores resultados, eran más rápidas o eran más fáciles de aprender y de utilizar.

4.3.1 ImageMagick

ImageMagick es un software de código abierto que soporta más de 100 formatos de imagen distintos y que nos permite mostrar, manipular y convertir imágenes.

El software consiste principalmente en un conjunto de utilidades que pueden ser ejecutadas por línea de comandos, ya que no cuenta con una interfaz gráfica de usuario propia como sí la tienen otros programas con utilidades muy similares como GIMP o Adobe Photoshop. Además también incluye una API para diversos lenguajes de programación, dicha API es la que nos interesa para nuestro proyecto y será la que utilizaremos para gran parte de los análisis que serán realizados.

Entre los lenguajes de programación que permite la API se encuentra PHP, lenguaje que utilizamos para el desarrollo de nuestro trabajo. Para utilizarla se requiere IMagick, una extensión nativa de PHP para crear y modificar imágenes utilizando la API de ImageMagick mencionada anteriormente.

Esta extensión es la que utilizaremos principalmente para obtener toda la información de la imagen para poder compararlos con las especificaciones que han sido introducidas previamente en nuestra base de datos por una editorial para una revista específica y poder informar al usuario de que requisitos cumple y cuáles deben ser modificados. Concretamente utilizaremos esta API para obtener el formato de la imagen, el tamaño en píxeles y pulgada, el tipo de imagen (color, blanco y negro...), cantidad de píxeles por pulgada (DPI) y el peso de la imagen.

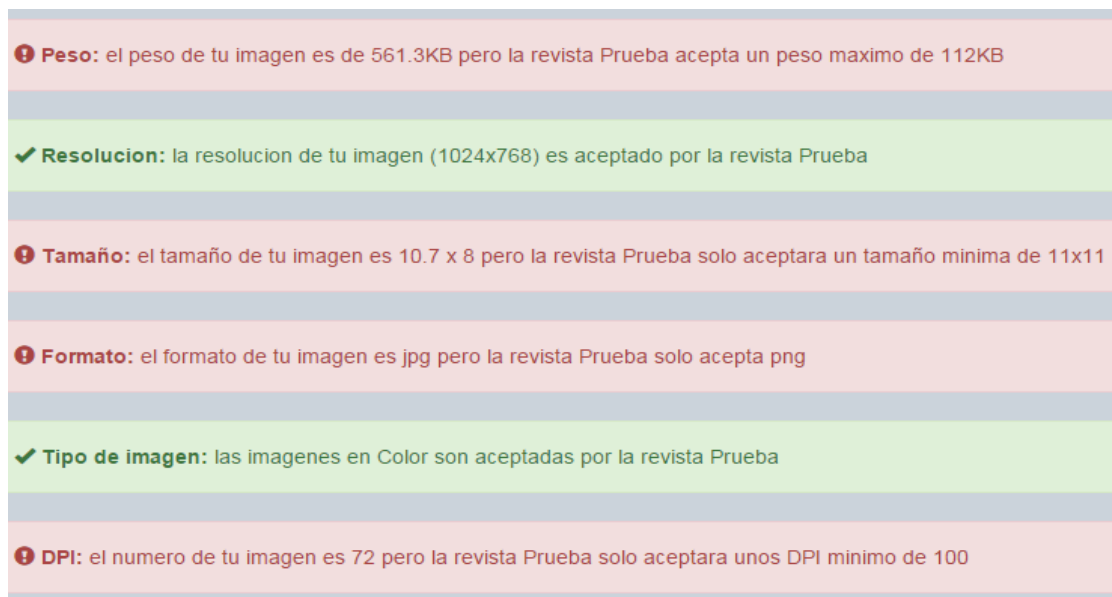


Figura 13. Ejemplo de resultados obtenidos con ImageMagick.

Como alternativa a ImageMagick existe la librería GD. Ambas son muy similares entre si, pero hemos considerado que era mejor selección ImageMagick por su mayor soporte de extensiones, mayor número de métodos implementados y permitiendo crear un código mas limpio.

4.3.2 ABBYY Cloud OCR

ABBYY es un proveedor de captura de documentos, ICR (intelligent character recognition), servicio de traducción de lenguas y de OCR. Su Cloud OCR SDK permite a los desarrolladores integrar la tecnología de reconocimiento óptico de caracteres de ABBYY para aplicaciones de todo tipo, como pueden ser de fotografía o de conversión de imágenes a texto. Con esta API, los usuarios pueden subir imágenes al servidor OCR, proceso que necesita unos parámetros de reconocimiento y de exportación, obteniendo la respuesta del proceso en un fichero XML.

Cloud OCR SDK es una API online para OCR, ICR, y otros procesos de reconocimiento de imagen. Permite a los desarrolladores utilizar su software para realizar OCR en sus aplicaciones con un bajo nivel de inversión. Además permite flexibilidad para escalar el proceso respecto a la demanda actual sin ningún tipo de inversión en hardware.

Concretamente esta API la utilizaremos para realizar el OCR del texto que puede haber en la imagen analizada. Este texto será mostrado al usuario para que lo puede utilizar en el texto alternativo.

En nuestro caso, al tratarse nuestra aplicación web de un prototipo no nos ha sido necesario comprar ninguno de los paquetes que se ofrecen para poder utilizar su API y ha sido suficiente con la utilización de la versión de prueba que proporcionan para desarrolladores. Los servicios que nos proporciona esta versión de prueba son:

- Soporte para reconocer 198 lenguas diferentes.
- Disponibilidad de 50 páginas para OCR.
- Reconocimiento de pequeños textos de un máximo de 50 caracteres en coordenadas concretas de la imagen.
- Reconocimiento de códigos de barras.
- Reconocimiento de escritura a mano.

Estos servicios han sido suficientes para el desarrollo de este proyecto, pero en caso de llegar a comercializarse se tendría que adquirir uno de los paquetes de pago o buscar una API diferente.

En la actualidad, el plan de pago más barato que ofrece esta API es de 10\$ y nos ofrece un máximo de 100 páginas para OCR y un coste adicional de 0.10\$ por página extra. Como alternativa existe otra API llamada OCR-IT, esta otra API nos permite pagar solo por lo que usemos, con un coste de 0.10\$ por página. Dado que con la segunda API solo pagamos por lo que usamos y que en los primeros pasos de nuestra aplicación es probable que no haya mucho tráfico probablemente fuese una mejor elección.

4.3.3 Imagga

Imagga es una plataforma de reconocimiento de imágenes que provee una API para etiquetar imágenes, disponible para desarrolladores y negocios.

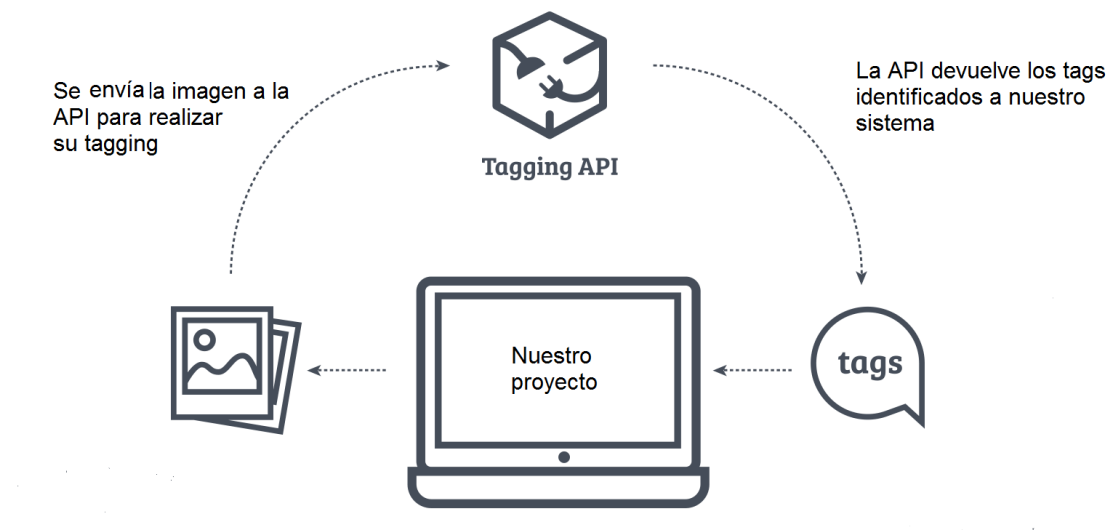


Figura 14. Ilustración del funcionamiento de Imagga. (Fuente: <https://imagga.com/>)

Imagga nos permite subir imágenes a su aplicación para su identificación. Como resultado devolverá una lista de etiquetas que clasificará nuestra imagen.

```
$cliente = new \Imagga\Imagga\Client($config['api_key'], $config['api_secret']);
$tags = $cliente->tagging($imagen,array("language" => "es"));
```

Figura 15. Ejemplo de conexión con Imagga.

✓Tags: , rock, cañón, paisaje, montaña, faro, valle, cielo, viajar, turismo, piedra

Figura 16. Ejemplo de resultado.

4.3.4 Cloudinary

Cloudinary es un servicio web para la gestión de imagen en páginas webs y en aplicaciones para móviles.

Cloudinary nos permite subir las imágenes a analizar a sus servidores permitiéndonos almacenar y manipular las imágenes de una forma más eficiente.

```
Cloudinary::config(array(
    "cloud_name" => "dnqs15j2j",
    "api_key" => "282356721283397",
    "api_secret" => "aPU4cp_v4T3dkrmvlf7eW4J8TdE"
));

$data = Cloudinary\Uploader::upload($imagen_a_subir);
$imagen = $data["url"];
```

Figura 17. Ejemplo de conexión con Cloudinary y subida de una imagen.

En este proyecto, utilizaremos esta API para almacenar las imágenes que los usuarios suban a nuestra aplicación durante el transcurso del análisis. Una vez finalizado eliminamos las imágenes de esta API.

4.4 Web Hosting

Un web hosting o alojamiento web es un servicio ofrecido por un proveedor que proporciona unos servidores conectados a internet donde pueden ser almacenados todo tipo de ficheros que podrán ser accedidos desde cualquier navegador web.

El uso más habitual que se le da a este tipo de servicio, y el que le vamos a dar en el desarrollo de esta aplicación, es el de alojamiento para páginas web. Esto se consigue subiendo al servidor proporcionado todos los archivos, librerías, imágenes y scripts, y permitiendo que nuestra página web pueda ser consultable en cualquier lugar del mundo.

En la actualidad hay un gran número de hosting disponibles que proporcionan una versión gratuita para que pueda ser usada durante el desarrollo de aplicaciones web. Estas versiones gratuitas suelen contar con una serie de limitaciones y no son aptas para alojar proyectos comerciales.

En nuestro caso, puesto que nuestro proyecto se trata de un prototipo, para comprobar la funcionalidad de nuestra aplicación nos ha sido suficiente con una de estas versiones gratuitas. En concreto usaremos OpenShift de Red Hat.

En caso de que el proyecto acabase siendo comercializado sería necesario contratar un plan de pago, ya sea utilizando este el mismo web hosting que para el desarrollo o buscando uno diferente.

4.4.1 OpenShift

OpenShift Online es una plataforma como servicio (platform as a service, PaaS) en la nube propiedad de Red Hat que nos permite tener nuestro propio web hosting para el alojamiento de nuestro proyecto de forma gratuita, incluyendo soporte para nuestra base de datos. Aparte del alojamiento de nuestro proyecto este modelo de servicio nos ofrece una plataforma con todas las herramientas necesarias para el desarrollo del proyecto con total

control sobre él, pero sin tener que ocuparnos de la infraestructura. Para acceder a la web de nuestro proyecto solo hay que acceder a este [enlace](#).

Esta versión gratuita nos proporciona:

- Un dominio gratuito. Este dominio puede ser modificado por cualquier otro que poseamos de forma gratuita.
- Capacidad de almacenaje de 1GB. Suficiente para el almacenaje de nuestro proyecto.
- Soporte de PHP.
- Posibilidad de añadir de una base de datos.
- Soporte de MySQL y el gestor de bases de datos PhpMyAdmin.
- Control de versión por Git en el propio servidor.

Todas estas características han sido las que nos han hecho inclinarnos por este alojamiento web.

4.4.2 Otros web hosting

Durante la búsqueda de un web hosting, antes de la elección de OpenShift, se barajaron otras posibilidades. En este apartado comentaremos algunos de estos web hostings alternativos y cuáles son sus principales características.

4.1.2.1 LAMP

Entre estas posibles alternativas destaca LAMP. LAMP es un acrónimo que deriva de los cuatro principales componentes que lo forman: el sistema operativo Linux, el servidor HTTP de Apache, el sistema MySQL de base de datos y por último PHP como lenguaje de programación.

La principal ventaja que nos proporciona LAMP es la posibilidad de crear un servidor virtual en nuestra máquina permitiéndonos simular como funcionaría nuestra aplicación web en un entorno real pero sin la necesidad de contar con un web hosting.

4.1.2.2 Azure

Azure es el nuevo servicio en la nube creado por Microsoft. Permite el desarrollo de aplicaciones para la web y móviles accesibles desde cualquier tipo de dispositivo. Su característica principal es que ha sido diseñado para la simplificación del código evitando la repetición de funciones. También permite ser integrado en sistemas Software como servicio

(SaaS por sus siglas en ingles) y empresariales y cumplir las necesidades de seguridad, confianza y escalabilidad.

Azure permite alojar diferentes tipos de aplicaciones:

- Aplicaciones web
- Aplicaciones móviles
- API
- Aplicaciones lógicas

En nuestro caso el tipo de aplicación que necesitaríamos seria del tipo web.

Al igual que OpenShift, Azure también proporciona una versión gratuita de su servicio para usar durante el desarrollo de la aplicación.

Las principales características que nos proporciona son:

- Capacidad de almacenaje de 1GB. Suficiente para el almacenaje de nuestro proyecto.
- Soporte de PHP y MySQL.
- Control de versión por Git y GitHub.
- Escalado automático.

4.1.3 ¿Por qué elegir OpenShift?

El primero de los web hostings anteriores que fue descartado fue LAMP, la principal razón fue que el resto de web hostings comparados nos proporcionan un servicio en la nube en el que no nos tenemos que preocupar por el mantenimiento de los servidores mientras que con LAMP corre de nuestra cuenta. Además tanto OpenShift como Azure permiten un acceso más fácil tanto para la tutora, los miembros del tribunal y para los usuarios que realizaran los test de accesibilidad, proporcionando un entorno más similar al que tendría el producto final.

Como se ha podido observar en el apartado anterior OpenShift y Azure son muy parecidos entre si y las características que nos proporcionan son prácticamente idénticas y ambas nos proporcionan todas las características que necesitamos para nuestro proyecto por lo que cualquiera de las dos podría haber sido una buena elección. La inclinación final por OpenShift ha sido por contar con su propio Git integrado.

4.5 Git

Git es un proyecto open source originalmente creado por Linus Torvalds, conocido por ser el desarrollador de Linux. Git es un sistema creado para realizar el control de versiones, permitiendo gestionar y almacenar diferentes versiones de un mismo proyecto de forma sencilla. En lugar de aplicar cambios directamente en el repositorio principal de un proyecto, como solían hacer proyectos anteriores a Git, este permite realizar una copia local en el sistema del usuario donde realizar todas las modificaciones necesarias. Después de realizar los cambios, el usuario puede subir al servidor principal las modificaciones necesarias. Gracias a este método se elimina la necesidad de tener que estar conectando con el servidor principal cada vez que se realice un cambio en el proyecto.

Para el desarrollo de este proyecto se ha utilizado el servicio de hosting de Git que viene incluido con el web hosting OpenShift que ha sido explicado previamente.

4.6 SublimeText

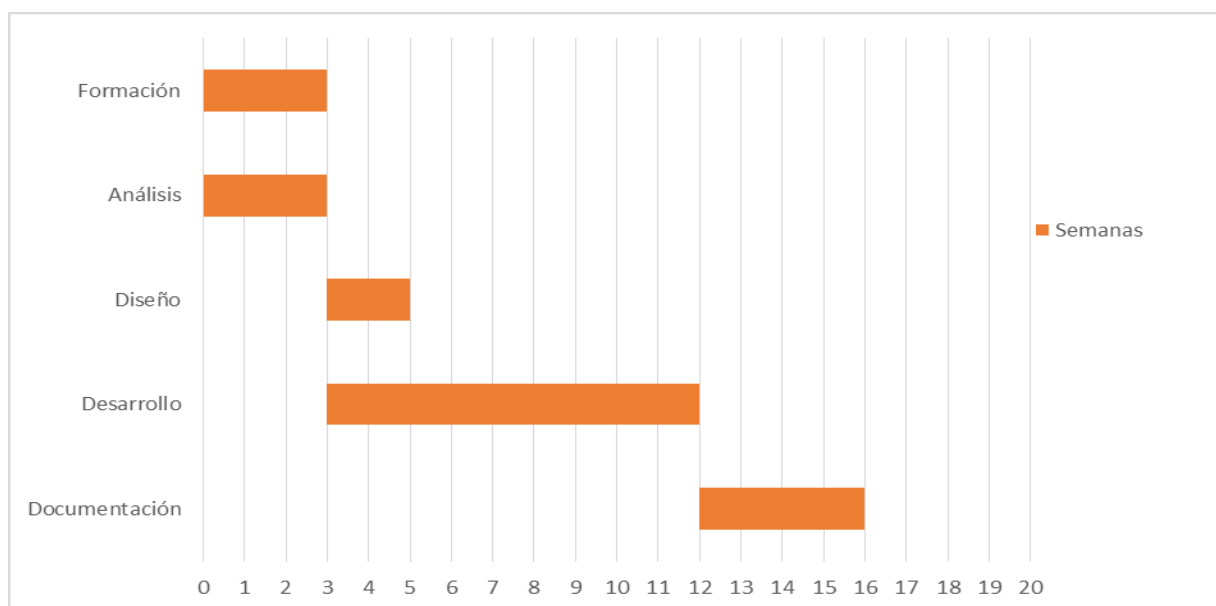
SublimeText es un editor de código y de texto. Es capaz de soportar gran cantidad de lenguajes de programación distintos, entre los que se incluyen todos los que han sido utilizados para el desarrollo de este trabajo. Nos ha parecido un editor muy completo y útil, por esta razón será el que utilizemos durante el desarrollo de la aplicación.

5. PLANIFICACIÓN

Durante la planificación de este proyecto se decidió dividir el proyecto en diferentes etapas.

- **Formación:** En esta etapa del proyecto se incluye todo el tiempo dedicado al aprendizaje para el correcto desarrollo del proyecto.
- **Análisis de requisitos:** Aquí está incluido todo el trabajo dedicado a investigar las funcionalidades necesarias a incluir en la aplicación.
- **Diseño:** En esta etapa se incluye el diseño de la base de datos y el diseño del front-end de la aplicación.
- **Desarrollo:** En esta etapa es donde contamos el tiempo invertido en la programación del código necesario para el proyecto y el tiempo dedicado a las pruebas de funcionamiento.
- **Documentación:** Esta etapa abarca la creación de toda la documentación necesaria para el proyecto.

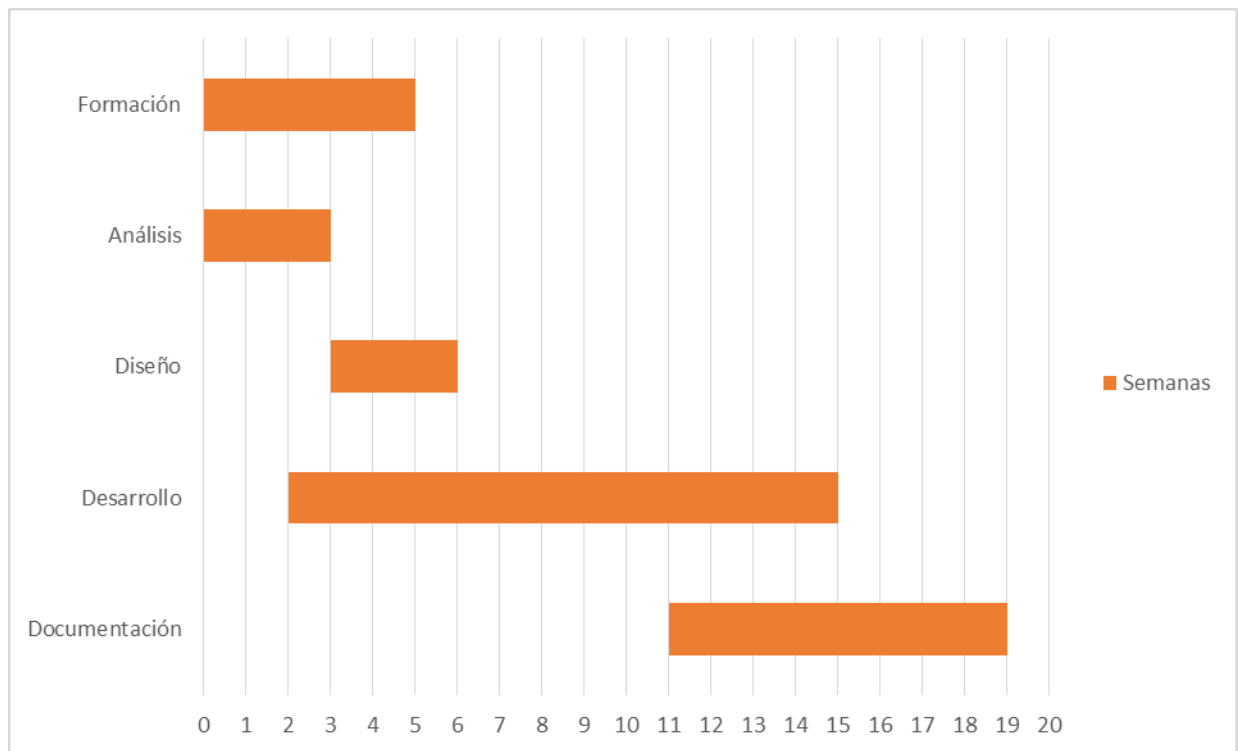
Para la división del tiempo se hizo una estimación inicial basándonos en estas cinco etapas y con un tiempo estimado para realizar el proyecto de 17 semanas (Septiembre-Diciembre), dando como resultado el siguiente diagrama de Gantt:



	Formación	Análisis	Diseño	Desarrollo	Documentación
Semana Inicio	0	0	3	3	11
Semanas	3	3	2	9	4

Figura 18. Diagrama de Gantt inicial.

Una vez finalizado el desarrollo del proyecto se volvió a realizar el diagrama de Gantt, pero ahora teniendo en cuenta las horas reales invertidas. Este es el resultado:



	Formación	Análisis	Diseño	Desarrollo	Documentación
Semana Inicio	0	0	3	2	11
Semanas	5	3	3	13	8

Figura 19. Diagrama de Gantt final.

Estos cambios respecto a la planificación inicial se han debido a que tanto a que en la etapa de formación, pero sobre todo, en la etapa de documentación se han necesitado mas horas de las previstas. Esto ha sido debido a una dificultad personal para la redacción de documentos.

6. PRUEBAS REALIZADAS

Durante el desarrollo del proyecto se han ido realizado diferentes tipos de pruebas de uso para verificar el correcto funcionamiento y desarrollo de la aplicación.

Una de estas pruebas de funcionamiento consistió en la utilización de la herramienta desarrollada por Google PageSpeed Insights creada para realizar un testeo sobre el rendimiento de la página web y sugerirnos ciertas mejoras que puedan mejorar la velocidad de carga de las páginas. Esta herramienta separa las sugerencias entre ordenadores y dispositivos con pantallas más pequeñas como móviles o tablets.

El resultado es el siguiente:

Resultados para móvil

82 / 100 Velocidad

! Elementos que puedes plantearte corregir:

Minificar JavaScript

► [Mostrar cómo corregirlo](#)

Eliminar el JavaScript que bloquea la visualización y el CSS del contenido de la mitad superior de la página

► [Mostrar cómo corregirlo](#)

Especificar caché de navegador

► [Mostrar cómo corregirlo](#)

✓ 7 reglas aprobadas

▼ [Ocultar detalles](#)

Evita los redireccionamientos a páginas de destino

Tu página no tiene redireccionamientos. Más información sobre [cómo evitar redireccionamientos de páginas de destino](#)

Habilitar compresión

La compresión está habilitada. Más información sobre la [habilitación de la compresión](#)

Minificar CSS

El CSS está reducido. Más información sobre la [reducción del CSS](#)

Minificar HTML

El HTML está reducido. Más información sobre la [reducción del HTML](#)

Optimizar imágenes

Tus imágenes están optimizadas. Más información sobre la [optimización de imágenes](#)

Prioriza el contenido visible

El contenido de la mitad superior de la página está correctamente priorizado. Más información sobre [la priorización de contenido visible](#)

Reducir el tiempo de respuesta del servidor

Tu servidor respondió rápidamente. Más información sobre la [optimización del tiempo de respuesta del servidor](#)

Figura 20. Resultados PageSpeed Insights para móvil.

Resultados para ordenador

82 / 100 Velocidad

! Elementos que puedes plantearte corregir:

Minificar JavaScript

► [Mostrar cómo corregirlo](#)

Eliminar el JavaScript que bloquea la visualización y el CSS del contenido de la mitad superior de la página

► [Mostrar cómo corregirlo](#)

Especificar caché de navegador

► [Mostrar cómo corregirlo](#)

✓ 7 reglas aprobadas

► [Mostrar detalles](#)

Figura 21. Resultados PageSpeed Insights para ordenador.

Además, se nos ha proporcionado una valoración de la experiencia de usuario.

100 / 100 Experiencia de usuario

✓ Enhorabuena. No se han encontrado problemas.

Adaptación del contenido a la ventana gráfica

El contenido de tu página encaja con la ventana gráfica. Más información sobre cómo [adaptar el contenido a la ventana gráfica](#)

Aplicar el tamaño adecuado a los botones táctiles

Todos los enlaces y los botones de la página son suficientemente grandes para que los usuarios los puedan tocar en una pantalla táctil. Más información sobre cómo puedes [aplicar el tamaño correcto a los botones táctiles](#)

Configurar ventana gráfica

Tu página especifica una ventana gráfica que se adapta al tamaño del dispositivo, lo que permite que se procese correctamente en todos los dispositivos. Más información sobre cómo [configurar ventanas gráficas](#).

Evita intersticiales de instalación de aplicaciones que oculten contenido

Parece que tu página no tiene ningún intersticial de instalación de aplicaciones que oculte una cantidad significativa de contenido. Puedes obtener más información acerca de la importancia de [evitar el uso de intersticiales de instalación de aplicaciones](#).

Evitar los plugins

Parece que tu página no utiliza plugins y esto podría impedir que el contenido se utilice en muchas plataformas. Puedes obtener más información acerca de la importancia de [evitar los plugins](#).

Utilizar tamaños de fuente que se puedan leer

El texto de la página es legible. Más información sobre cómo [utilizar tamaños de fuente que se puedan leer](#).

Figura 22. Resultados PageSpeed Insights de la experiencia de usuario.

Como se puede observar en las imágenes anteriores, no se ha encontrado ningún error grave que deba ser corregido, pero si se nos sugieren diferentes modificaciones para mejorar el funcionamiento de la aplicación. Las sugerencias que nos devuelve son para realizar la compresión de JavaScript y para el uso de la caché del navegador, pero se han descartado por no considerarse necesarios para nuestra aplicación actual, aunque muy probablemente se tenga que tener en cuenta en un futuro si la aplicación crece.

6.1 Test de usuarios

Para comprobar el funcionamiento real de nuestra aplicación web y comprobar si había algún tipo de problema de usabilidad se han realizado diversos test de usuario simulando ser usuarios finales de nuestra página.

Lo primero que se ha hecho antes de poder realizar estos test ha sido distinguir los diferentes tipos de usuarios a los que esta dirigida nuestra aplicación. En este caso se han encontrado dos tipos de usuarios, autores y editores.

- **Autores:** Este tipo de usuario será el que utilizará la web para subir las imágenes que utilizará en su artículo para comprobar su accesibilidad y crear un texto alternativo para cada una de dichas imágenes.
- **Editores:** Serán los encargados de añadir la información sobre los requisitos mínimos de accesibilidad que deberán cumplir las imágenes de los artículos que vayan a ser publicados en su revista.

Una vez que ya tenemos los tipos de usuario definidos diseñamos una tarea específica para cada uno de ellos. El resultado obtenido de la realización de estas tareas será lo que nos indique si la aplicación ha superado los test de usuario o no.

Las tareas que se les ha asignado a cada uno de los usuarios, teniendo en cuenta su tipo de usuario, han sido las siguientes:

- **Tarea usuario autor:** Eres el autor de un nuevo artículo para la revista RevistaTfg. En este artículo que estás redactando quieres incluir una imagen para ilustrar el contenido. Para poder publicar esta imagen en la revista antes debes comprobar que cumple los requisitos mínimos para la revista. Para poder comprobarlo con facilidad y rapidez has decidido utilizar nuestra aplicación web.

Lo primero que deberás hacer es acceder a la página web donde deberás seleccionar tu imagen, la revista y proceder a realizar su análisis.

Deberás comprobar que la imagen cumple todos los requisitos mínimos, si es así, procederás a la creación del texto alternativo para la imagen.

Una vez finalices el texto alternativo tendrás que descargarlo en tu ordenador para finalizar la tarea.

- **Tarea usuario editor:** Eres el editor de una revista, para poder cumplir con la legislación vigente necesitas que todas las imágenes que se publiquen en tu revista cumplan unos requisitos mínimos para ser consideradas accesibles. Para facilitar que los autores puedan comprobar con facilidad la accesibilidad de sus imágenes has decidido añadir la información de tu revista a nuestra página web.
Para realizar esta tarea deberás acceder a la página web y rellenar el formulario con la información requerida, una vez finalizado deberás subir esta información para que los autores puedan acceder a ella.

La información que debes añadir es la siguiente:

- Nombre: tu nombre
- Correo: revista@gmail.com
- Nombre revista: inventado
- Formatos aceptados: .png, .bmp
- Tipos de imagen y DPI(puntos por pulgada): color->72
- Resolución: 1024x 720 pixeles
- Tamaño: 12x12 inches(pulgadas)
- Peso: 2000 KB
- Número palabras en leyenda: 10 palabras
- Número de palabras en el título: 50 palabras

Estos son los distintos usuarios con los que se han realizado los test:

- Usuario 1:
 - Fecha realización: 15/01/16
 - Duración: 6 minutos
 - Formación: Estudiante de química
 - Edad: 23

- Usuario 2:
 - Fecha realización: 15/01/16
 - Duración: 10 minutos
 - Formación: Estudiante de filosofía
 - Edad: 24
- Usuario 3:
 - Fecha realización: 16/01/16
 - Duración: 4 minutos
 - Formación: Educador social
 - Edad: 26
- Usuario 4:
 - Fecha realización: 16/01/16
 - Duración: 5 minutos
 - Formación: Fisioterapeuta
 - Edad: 24
- Usuario 5:
 - Fecha realización: 17/01/16
 - Duración: 9 minutos
 - Formación: Trabaja en marketing online
 - Edad: 27

El documento de presentación de la tarea a los usuarios se puede ver en el Anexo III y los documentos de consentimiento se pueden ver en el Anexo IV.

6.1.1 Resultados de los test

En este apartado explicaremos los resultados que se han obtenido de los diferentes test que se han realizado diferenciando entre los dos tipos de usuarios, los autores y los editores.

6.1.1.1 Autores

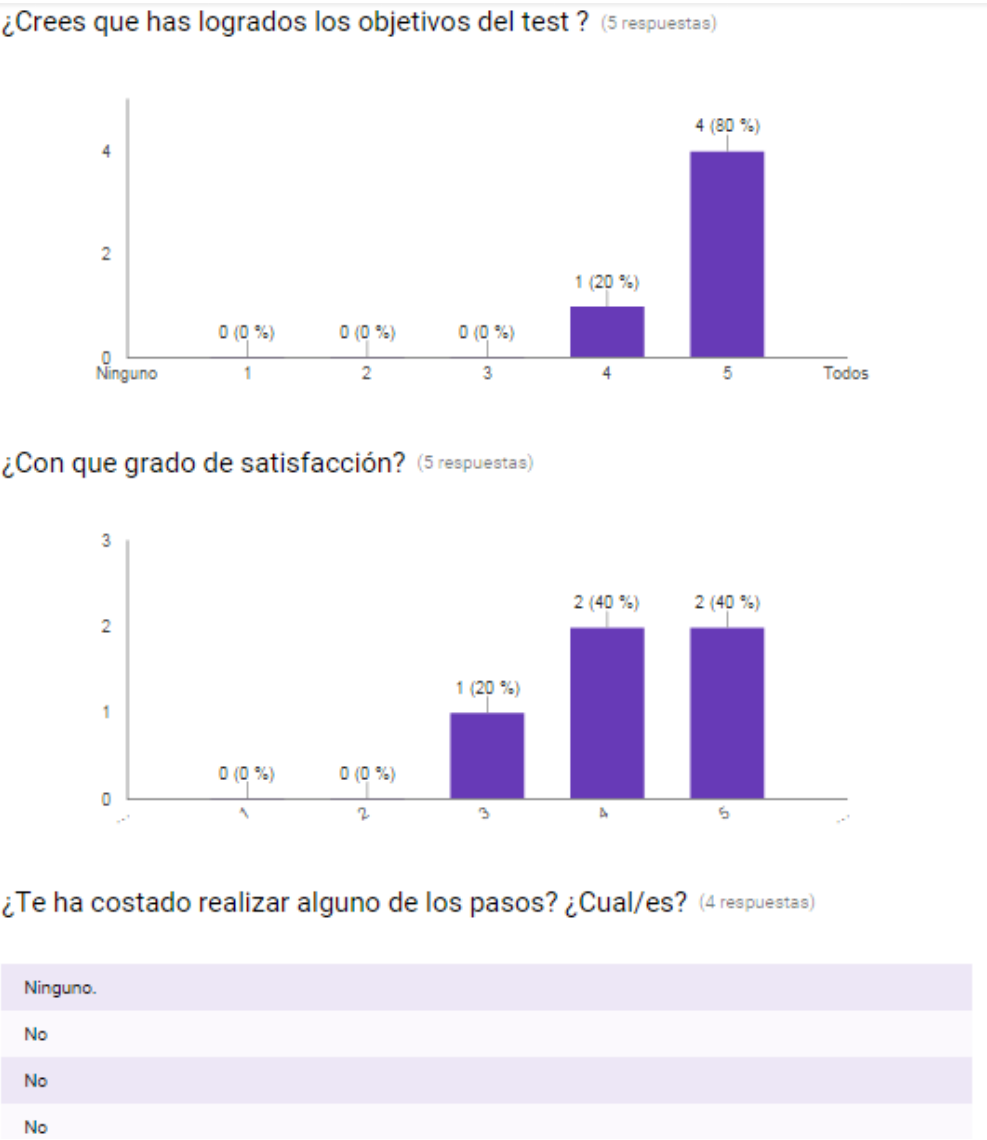
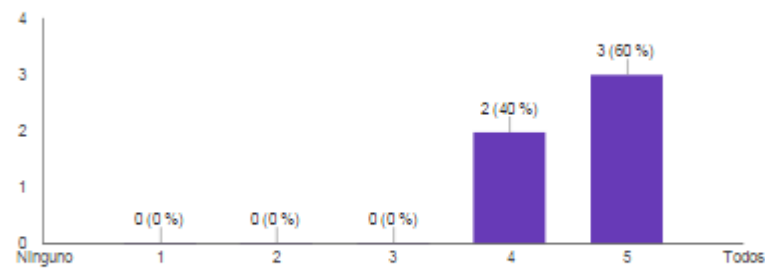


Figura 23. Ilustración de las respuestas 1, 2 y 3.

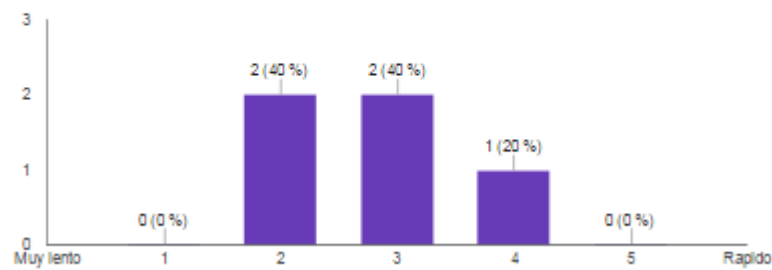
¿Has podido comprender con facilidad los resultados del análisis? (5 respuestas)



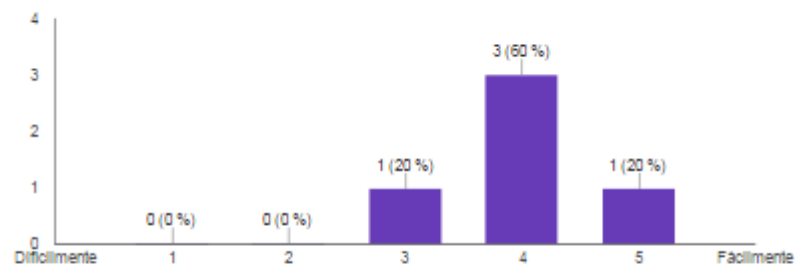
¿Si no has comprendido alguno, cual? (0 respuestas)

Aún no hay respuestas para esta pregunta.

¿Consideras que el tiempo que ha tardado el análisis ha sido lento? (5 respuestas)



¿Has podido comprobar que tu imagen podía ser visualizada, sin errores, por un daltónico? (5 respuestas)

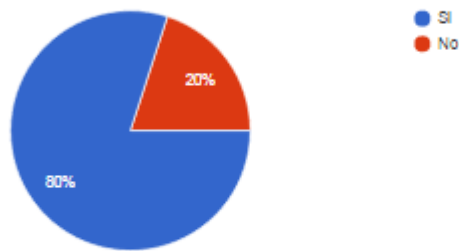


¿En caso negativo, porqué? (0 respuestas)

Aún no hay respuestas para esta pregunta.

Figura 24. Ilustración de las respuestas 4, 5, 6, 7 y 8.

¿Sobre el formulario para el texto alternativo, has podido realizarlo sin problemas?
(5 respuestas)



¿Has comprendido todas las preguntas que se te han realizado durante el formulario para el texto alternativo?
(5 respuestas)



Me gustaría comentar: (4 respuestas)

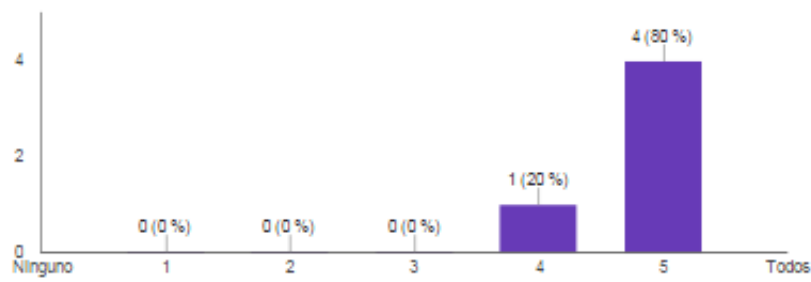
- No se pueden eliminar filas y columnas al crear la tabla en el formulario y no se puede retroceder.
- Estaría bien que desde la misma pagina se pudiese modificar la imagen para que cumpla todos los criterios.
- No se puede volver atrás en el formulario, si te equivocas hay que volver a analizar la imagen.
- Falta un botón para volver atrás en el formulario

Figura 25. Ilustración de las respuestas 9, 10 y 11.

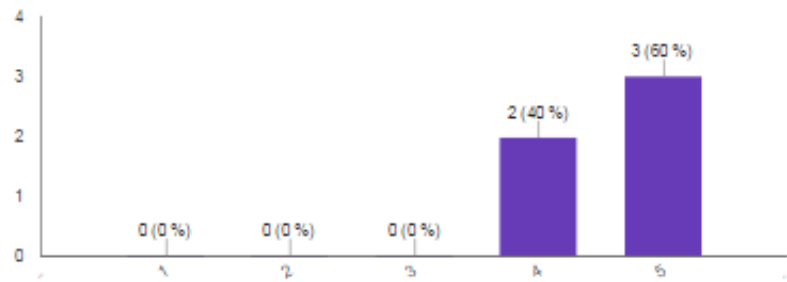
Las conclusiones que se han sacado es que los resultados que muestra la aplicación son fáciles de entender y los usuarios están satisfechos con el funcionamiento general de la página. La mayoría cree que la velocidad del análisis es algo lenta por lo que se tendrá que tener en cuenta en futuras iteraciones del proyecto. Finalmente los comentarios nos ayudaran a mejorar el funcionamiento y nos proporcionan ideas para futuras funcionalidades que serán comentadas en el apartado 7.1 Futuro. Comentar que en caso de seguir con el desarrollo del futuro, sería recomendable dar un paso atrás en el desarrollo de la aplicación para realizar una evaluación heurística y de esta forma conseguir un mejor diseño para la aplicación.

6.1.1.2 Editores

¿Crees que has logrado los objetivos del test? (5 respuestas)



¿Con que grado de satisfacción? (5 respuestas)

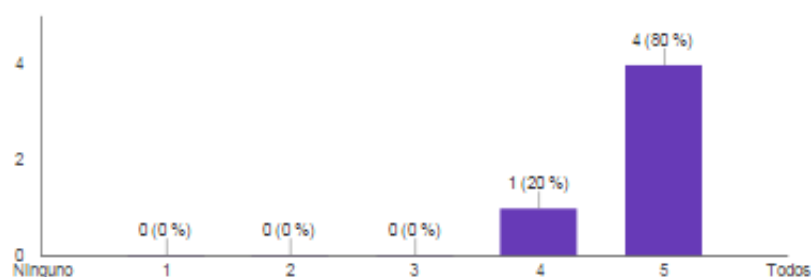


Si has tenido algún problema al realizar alguno de los pasos comenta cual ha sido.
(0 respuestas)

Aún no hay respuestas para esta pregunta.

Figura 26. Ilustración de las respuestas 1, 2 y 3.

¿Has podido entender que eran todos los datos que se pedían en el formulario?
(5 respuestas)



¿En caso de no haber entendido alguno, cuál ha sido? (0 respuestas)

Aún no hay respuestas para esta pregunta.

Me gustaría comentar: (4 respuestas)

Cambiaría el diseño del formulario.
El apartado nombre no permite mas de una palabra.
Estaría bien que se mostrasen mensajes de ayuda.
Nada.

Figura 27. Ilustración de las respuestas 4, 5 y 6.

Las conclusiones que se han sacado en el caso de los editores es que los usuarios están satisfechos con cómo funciona esta parte de la aplicación. Por lo tanto la experiencia de los usuarios ha sido positiva y los comentarios que nos han dado nos ayudarán a la mejora y evolución del proyecto.

7. CONCLUSIONES

Con el desarrollo de este proyecto hemos buscado la implantación de una aplicación web donde cualquier tipo de revista o editorial pueda indicar cuales son los requisitos mínimos que deben cumplir las imágenes que vayan a ser publicadas en los artículos de las revistas, y donde el autor pueda subir sus imágenes para ser analizadas y comprobar si cumplen los requisitos indicados por la editora y si dicha imagen es accesible para personas con problemas visuales. Además la aplicación permite que se les proporcione a los autores un formulario para crear un texto alternativo con la información que se muestra en la imagen.

En la actualidad este proyecto se encuentra en una fase de prototipo para comprobar su viabilidad como producto final. Esto quiere decir, que ha sido desarrollado para comprobar si los usuarios a los que va dirigidos estarían interesados en una aplicación con las características que se han descrito anteriormente. Esto implica que algunas de las características indispensables que tendría que tener como producto comercializable no han sido añadidas porque no eran necesarias en esta fase de prototipo. Teniendo en cuenta esto, consideramos que se han logrado todos los objetivos presentados durante el planteamiento del proyecto y se han logrado crear las bases necesarias para comprobar si este proyecto podría acabar siendo de utilidad para los autores y editores y si en caso de acabar desarrollándose un producto final lo utilizarían o no. Además se han creado unas bases que permiten poder seguir añadiendo funcionalidades para que el proyecto crezca y pueda acabar convirtiéndose en un producto final viable.

7.1 Futuro

En caso de que el proyecto se acabe formalizando se tendrían que añadir ciertas funcionalidades para poder ser lanzado al mercado. A continuación se explicarán cada una de ellas y también algunas funcionalidades extras que podrían ser añadidas.

- **Registro y login:** Como parte fundamental la aplicación deberá tener un sistema de login/registro para controlar el acceso de los usuarios a la información y las distintas funcionalidades de las que consta la web.

- **Modificación de revistas:** Con el sistema de registro se podría implementar una funcionalidad para que los editores puedan modificar la información que habían añadido previamente.
- **Reducir tiempo de análisis:** Actualmente los análisis de las imágenes de gran tamaño pueden tardar incluso minutos en analizarse, esto hace que el proceso pueda hacerse tedioso para los usuarios. Tendría que desarrollarse un nuevo método más eficiente que el creado para este prototipo.
- **Comprobación automática del contraste:** Actualmente el contraste de colores entre el texto que hay en la imagen y el color de fondo debe ser realizado de manera manual por el usuario con la herramienta proporcionada en la misma web. Sería deseable que fuese la misma aplicación la que realizase esta comparación de forma directa al analizar la imagen.
- **Comprobación automática de accesibilidad para usuarios con daltonismo:** En este prototipo se le muestran al usuario ejemplos de cómo visualizaría su imagen una persona con daltonismo, concretamente los tipos más comunes (tritanopia, protanopia, deuteranopia, acromatopsia) y debe ser el usuario el que decida si esa imagen es accesible para las personas con ese tipo de daltonismo o no. En un futuro sería recomendable que fuese la misma aplicación la que decida, mediante unos valores de calidad, si la imagen evaluada presenta problemas para este colectivo o no.
- **Mejorar el diseño del texto alternativo:** El diseño del formulario para el texto alternativo actual no acaba de convencer del todo, ya que algunos de los usuarios que han realizado los test no han quedado muy convencidos con su funcionamiento. Sería recomendable realizar un rediseño más agradable de utilizar, con más funcionalidades para poder moverse por el formulario, mostrar las respuestas seleccionadas, dar más posibilidades en la creación de tablas, etc.
- **Análisis múltiple de imágenes:** Ahora la aplicación permite analizar las imágenes una a una. Como mejora estaría bien que se pudiesen analizar múltiples imágenes a la vez. El inconveniente que supone esta mejora es que el tiempo de análisis actual

es algo elevado por lo que antes de aplicar esta mejora a la aplicación se tendría que realizar la explicada anteriormente de reducir los tiempos de análisis.

- **Registro de análisis:** Dado que en el futuro de esta aplicación contaríamos con un sistema de identificación de usuarios sería recomendable que los autores pudiesen acceder a un registro de todas las imágenes que han analizado para sus artículos.
- **Mejorar análisis OCR:** El análisis OCR que se realiza en este prototipo, aunque funciona y es capaz de detectar los textos en la imagen, da problemas según como esté posicionado el texto en la imagen. Esto hace que según como este colocado el texto en la imagen el análisis de OCR mezcle diferentes bloques de texto que no tienen relación directa, causando que el texto mostrado al usuario sea poco o nada comprensible. La mejora consistiría en que el usuario pudiese seleccionar estos bloques de texto en la imagen de forma manual (o que fuesen detectados por la aplicación) y se realizasen los análisis de OCR de forma individual a cada uno de estos bloques de imagen.
- **Modificar imágenes desde la aplicación:** Con el diseño actual, si los autores encuentran que su imagen no supera alguno de los requisitos mínimos marcados por el editor deben utilizar cualquier otra aplicación que les permita hacer las modificaciones necesarias. Para facilitar esta labor, se podrían añadir funcionalidades para que estas modificaciones se puedan realizar desde nuestra aplicación.

8. BIBLIOGRAFÍA

[1] OpenShift

<https://developers.openshift.com/en/managing-log-files.html>

[2] Guia Bootstrap

<http://getbootstrap.com/css/>

[3] Tutorial HTML5

<http://www.w3schools.com/html/>

[4] Guia de comandos de Git

<https://elbauldelprogramador.com/mini-tutorial-y-chuleta-de-comandos-git/>

[5] Manual de PHP

<http://www.php.net/manual/es/index.php>

[6] Manual PHP para ImageMagick

<http://php.net/manual/es/book.imagick.php>

[7] Manual MySQL

<http://www.php.net/manual/es/set.mysqlinfo.php>

<http://www.w3schools.com/sql/default.asp>

[8] Microsoft Azure

<https://azure.microsoft.com/es-es/services/app-service/web/>

[9] Documentación Imagga

<http://docs.imagga.com/>

[10] Documentación ABBY Cloud OCR

<http://ocrsdk.com/documentation/>

[11] Documentación OCR

<http://www.mattmahoney.net/ocr/>

[12] Tutorial JavaScript

<http://www.w3schools.com/js/default.asp>

[13] Documentación Cloudinary

http://cloudinary.com/documentation/php_integration

[14] Diseño responsivo

http://www.w3schools.com/html/html_responsive.asp

[15] Daltonismo

<https://es.wikipedia.org/wiki/Daltonismo>

<http://www.vischeck.com/daltonize/>

[16] Accesibilidad web

<http://accesibilidadweb.dlsi.ua.es/?menu=espanola>

[17] Legislación española

<http://olgacarreras.blogspot.com.es/2005/01/referencia-sobre-legislacin-espaola.html>

[18] Depósito digital de la UB

<http://diposit.ub.edu/dspace/handle/2445/34465/simplesearch?query=web>

[19] Pérdida de contraste

<http://visionsimulations.com/simulators/loss-of-contrast-simulator>

9. ANEXO

9.1 Anexo I: Contenido del ZIP

El fichero comprimido en .zip contiene dos ficheros. El primero es una carpeta con el nombre de tfgmanuel donde se encuentra toda la aplicación web que hemos desarrollado. El segundo fichero contiene la base de datos del proyecto.

La carpeta tfgmanuel está dividida en subcarpetas que contienen diversos archivos del proyecto de una forma más ordenada. En este anexo explicaremos brevemente que contiene cada una de estas carpetas y alguno de los ficheros más relevantes.

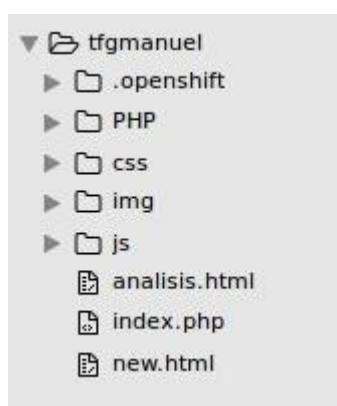


Figura 28. Contenido de la carpeta principal del proyecto.

La carpeta principal contiene las páginas web que serán mostradas al usuario y las subcarpetas para el resto de ficheros.

- **index.php:** El fichero principal de la página.
- **análisis.html:** Página que se mostrará para realizar el análisis y ver sus resultados.
- **new.html:** Contiene el formulario para añadir nuevas revistas a nuestra aplicación

En la carpeta PHP contiene los ficheros donde se realizan las funcionalidades principales de la aplicación. También contiene una sub carpeta llamada lib donde se encuentran las librerías de Imagga y Cloudinary.

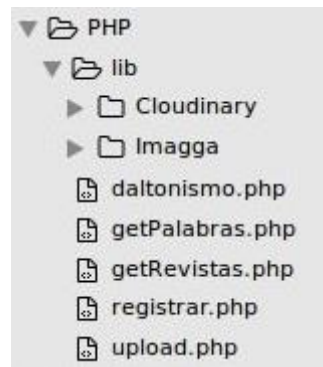


Figura 29. Contenido de la carpeta PHP.

Los ficheros principales son:

- **upload.php:** Lugar donde se realizará el análisis principal de la imagen.
- **registrar.php:** Aquí se añadirá la información de la nueva revista a nuestra base de datos.
- **daltonismo.php:** Se realizan las simulaciones para comprobar como sería vista la imagen por una persona daltónica.

La carpeta CSS contiene toda la información de los estilos utilizados en la página web.

La carpeta js contiene todo el contenido JavaScript. El código JavaScript está dividido en dos ficheros.

- **script.js:** Contiene todo el código principal de los scripts.
- **scriptForm.js:** Contiene los scripts específicos que se utilizan para subir la información de las revistas.

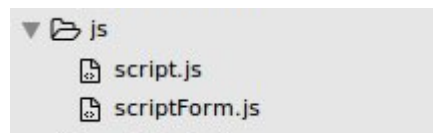


Figura 30. Contenido de la carpeta js.

Por ultimo en la carpeta img podemos encontrar las imágenes que se utilizan en la aplicación.

9.2 Anexo II: Manual de usuario

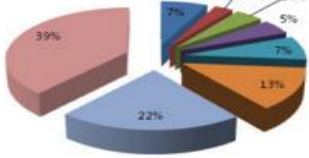
En este segundo anexo se describe el manual de cómo utilizar la aplicación. Para poder empezar lo primero que hay que hacer es acceder a la aplicación accediendo a este [enlace](#) mediante un navegador web, como por ejemplo Google Chrome.

9.2.1 Analizar revista

Para analizar una imagen lo primero que hay que hacer es seleccionar la imagen deseada y escribir el título y la leyenda de esta. Una vez realizado, se apretará el botón “Analizar” para comenzar el análisis.

Selección una imagen:

Seleccionar archivo castillog4.jpg



Publicación	Porcentaje
Revista española de documentación científica	39%
Anales de documentación: Revista de biblioteconomía y documentación	22%
Revista general de información y documentación	13%
Documentación de las ciencias de la información	7%
Scire: Representación y organización del conocimiento	5%
El profesional de la información	4%
Cuadernos de documentación multimedia	3%
Resto de publicaciones	7%

Título de la imagen

Introduce aquí la leyenda de la imagen.

Test

Analizar

Figura 31. Ejemplo de la introducción de los datos para el análisis.

Mientras se realiza el análisis de la imagen aparecerá un símbolo de carga. Dependiendo del tamaño de la imagen el análisis puede durar unos minutos.

9.2.2 Comprobar resultados

Una vez se ha completado el análisis, aparecerán en pantalla los resultados que se han obtenido.

Como se puede observar en la siguiente imagen, los datos que han pasado el test son marcados en color verde, mientras que los que no cumplen los requisitos mínimos son marcados en rojo. Para que la imagen sea considerada accesible por los editores de la revista el autor deberá modificar la imagen para que cumpla todos los requisitos.

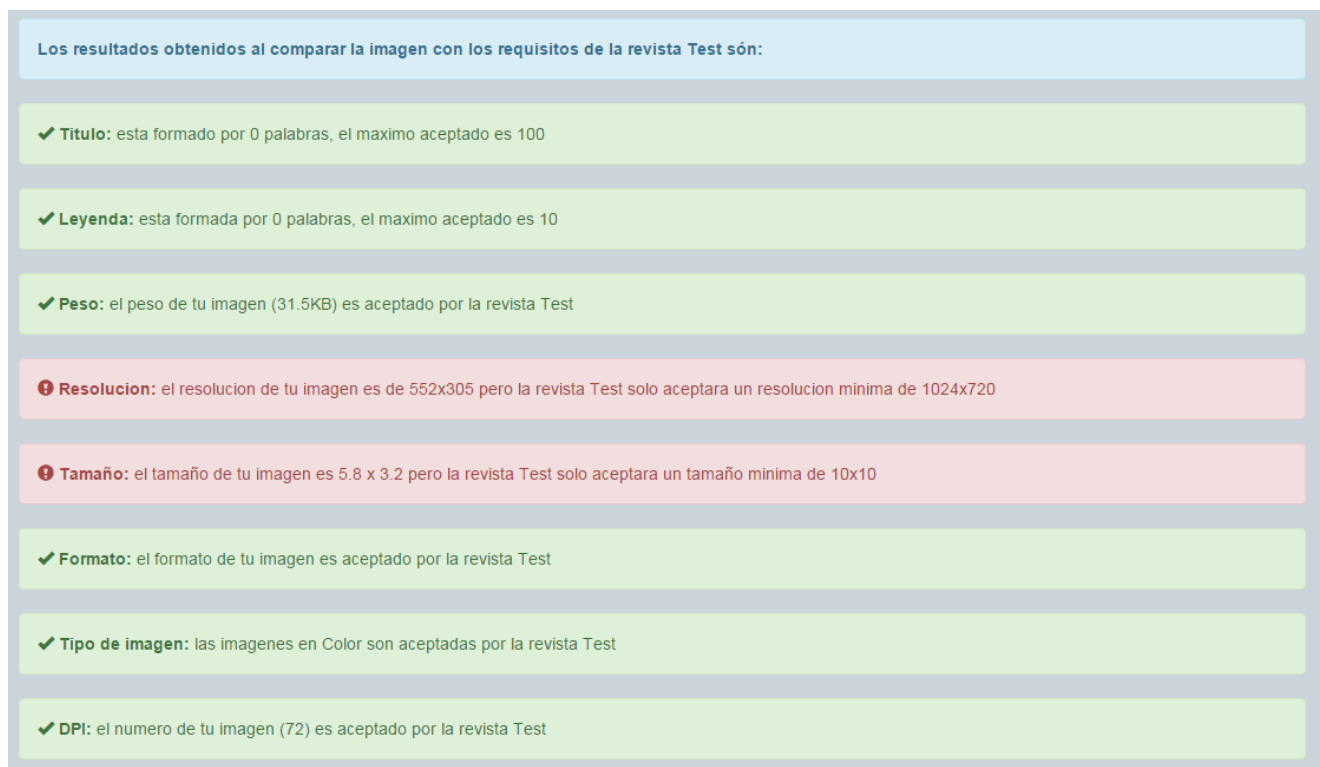


Figura 32. Ejemplo de resultado de un análisis.

En esta segunda imagen se pueden observar los resultados de realizar el OCR de la imagen y su tagging. Esta información no influye para decidir si la imagen es accesible o no, sino que es mostrada para que el autor pueda utilizar esta información para posteriormente crear el formulario alternativo.

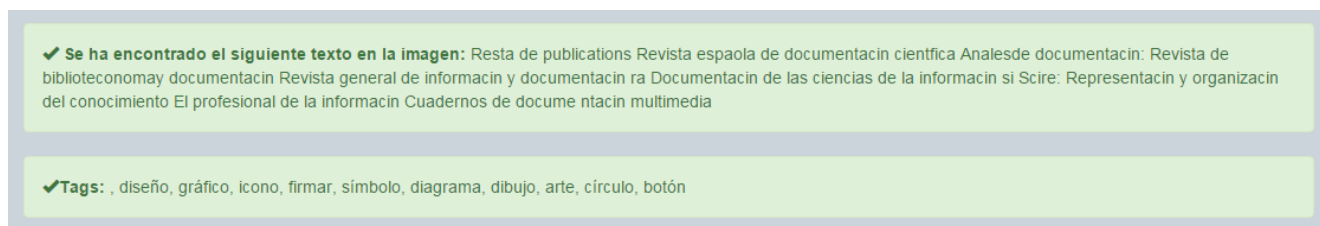


Figura 33. Resultado OCR y tags.

En la próxima imagen se muestra el resultado de los colores predominantes que se han encontrado en la imagen. Estos colores pueden ser utilizados en la herramienta que se muestra justo debajo para comprobar la calidad del contraste entre colores que aparezcan superpuestos.



Figura 34. Ejemplo de los colores predominantes en la imagen y herramienta para comprobar el contraste.

En esta última imagen de los resultados se puede ver la simulación de como vería nuestra imagen una persona con diferentes tipos de daltonismo. Se puede observar como una acromatopsia no sería capaz (o le sería muy complicado) distinguir a que representa cada una de las porciones del gráfico. Esto implica que el autor tendría que modificar la imagen añadiendo un nuevo tipo de simbología, aparte del color, para representar cada una de las porciones.

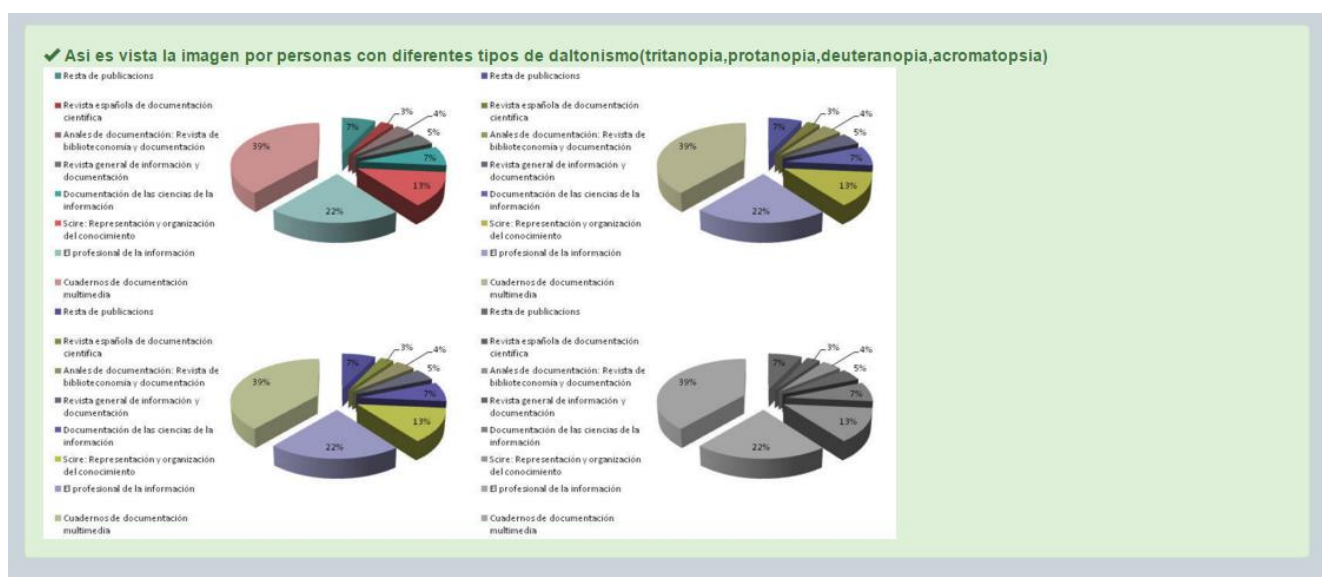
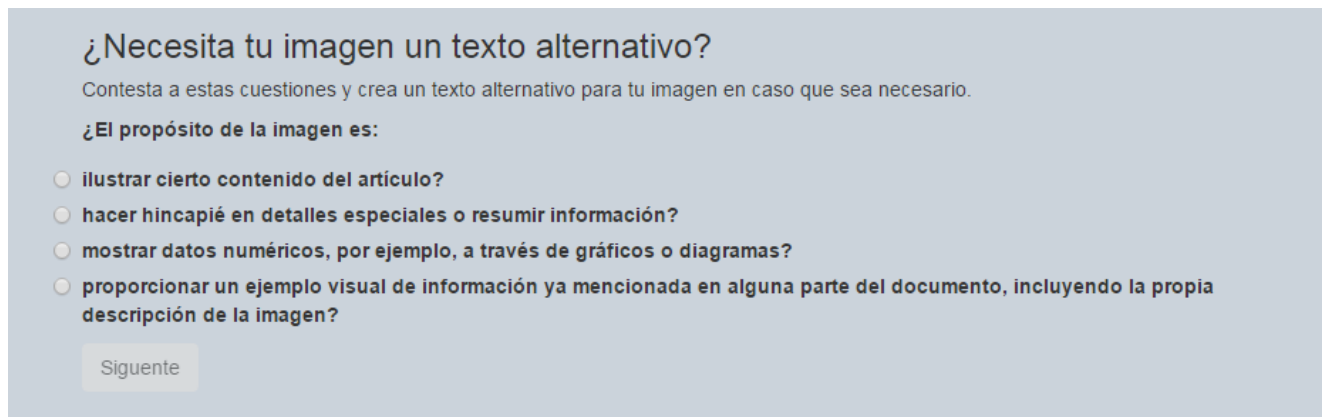


Figura 35. Ejemplo de resultado para la simulación del daltonismo.

9.2.3 Crear texto alternativo

Para saber qué información debe ser indicada en el texto alternativo el usuario debe ir contestando las preguntas que se le vayan realizando durante las distintas partes del formulario. Una vez terminado se le indicará la información que debe añadir y se le proporcionara las herramientas para crearlo en la misma aplicación. Una vez hecho esto se le dará la opción de descargar el texto alternativo.

El formulario tiene un título "¿Necesita tu imagen un texto alternativo?" y una instrucción: "Contesta a estas cuestiones y crea un texto alternativo para tu imagen en caso que sea necesario." La pregunta principal es "¿El propósito de la imagen es:". Hay cuatro opciones de radio: "ilustrar cierto contenido del artículo?", "hacer hincapié en detalles especiales o resumir información?", "mostrar datos numéricos, por ejemplo, a través de gráficos o diagramas?", y "proporcionar un ejemplo visual de información ya mencionada en alguna parte del documento, incluyendo la propia descripción de la imagen?". Al final hay un botón "Siguiete".

¿Necesita tu imagen un texto alternativo?

Contesta a estas cuestiones y crea un texto alternativo para tu imagen en caso que sea necesario.

¿El propósito de la imagen es:

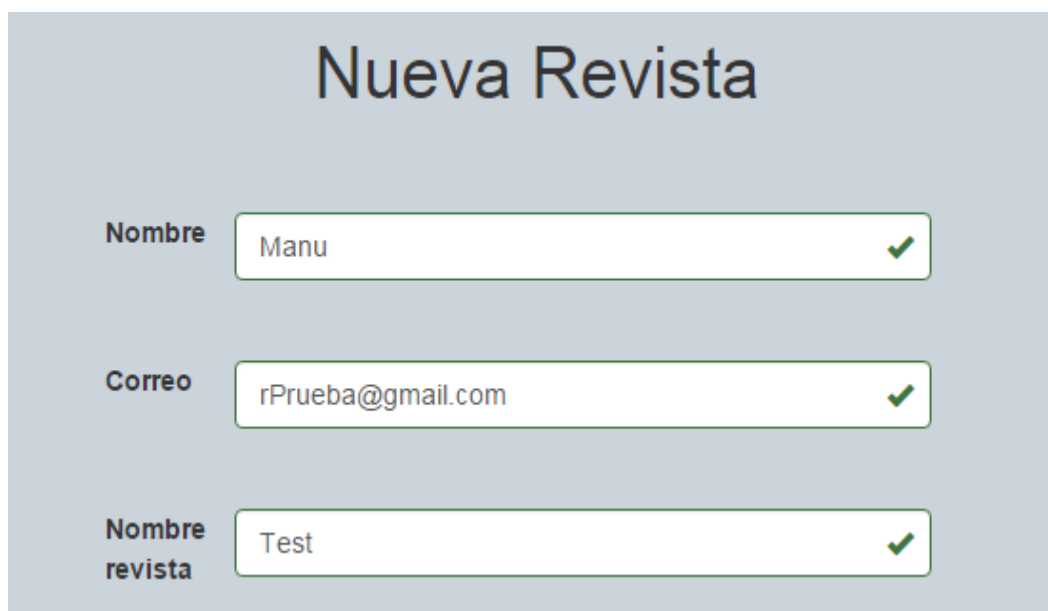
- ☐ ilustrar cierto contenido del artículo?
- ☐ hacer hincapié en detalles especiales o resumir información?
- ☐ mostrar datos numéricos, por ejemplo, a través de gráficos o diagramas?
- ☐ proporcionar un ejemplo visual de información ya mencionada en alguna parte del documento, incluyendo la propia descripción de la imagen?

Siguiete

Figura 36. Ilustración del formulario para el texto alternativo.

9.2.4 Añadir revista

Lo primero que se pedirá al editor para añadir una revista será una serie de datos para identificarse y el nombre de la revista.

El formulario se titula "Nueva Revista". Tiene tres campos de entrada: "Nombre" con el valor "Manu", "Correo" con el valor "rPrueba@gmail.com", y "Nombre revista" con el valor "Test". Cada campo tiene un icono de checkmark verde a la derecha.

Nueva Revista

Nombre

Correo

Nombre revista

Figura 37. Datos requeridos del editor.

En las siguientes dos imagen se muestra un ejemplo con toda la información que es requerida para añadir una revista y que deberá ser rellenada.

Formatos aceptados:

☐ jpg/jpeg
 ☒ png
 ☐ bmp
 ☐ gif
 ☐ tiff

Tipos de imagen y PPP (puntos por pulgada) mínimos:

☒ Color	72	✓
☒ Escala de grises	100	✓
☒ Blanco y negro	200	✓
☐ Line art	1200	

Figura 38. Selección de formatos, tipos de imagen y DPI.

Resolución mínima (píxeles):

✓ x ✓

Tamaño mínimo (Inch/Pulgadas):

✓ x ✓

Peso máximo:

✓ KB

Máximo numero de palabras en la leyenda:

✓

Máximo numero de palabras en el titulo:

✓

Enviar

Figura 39. Selección de resolución, tamaño, peso y numero maximo de palabras en la leyenda y el titulo.

En el caso de que se detecte algún tipo de incongruencia ente la información indicada en los apartados anteriores se mostraran diferentes tipos de alerta para advertir al usuario. Será este el que deberá decidir si los datos son realmente correctos e ignorar los mensajes de alerta o hacerles caso y hacer los cambios pertinentes.

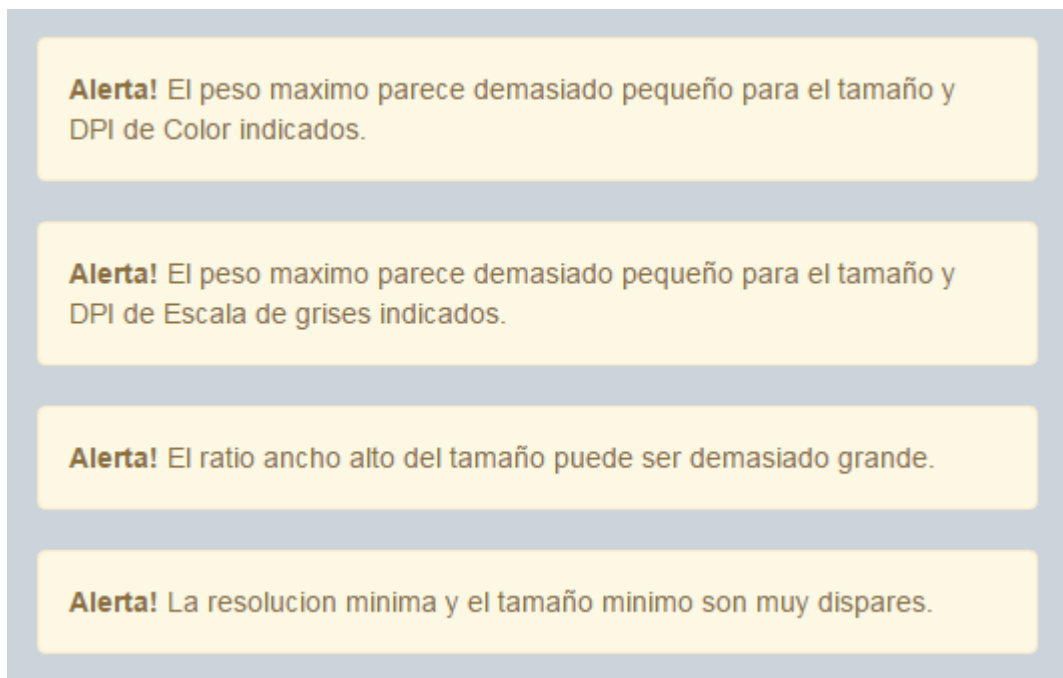


Figura 40. Ejemplos de mensajes de alerta.

Una vez la información de la revistas este almacenada para su posterior uso en nuestra aplicación se mostrará el siguiente mensaje de confirmación.

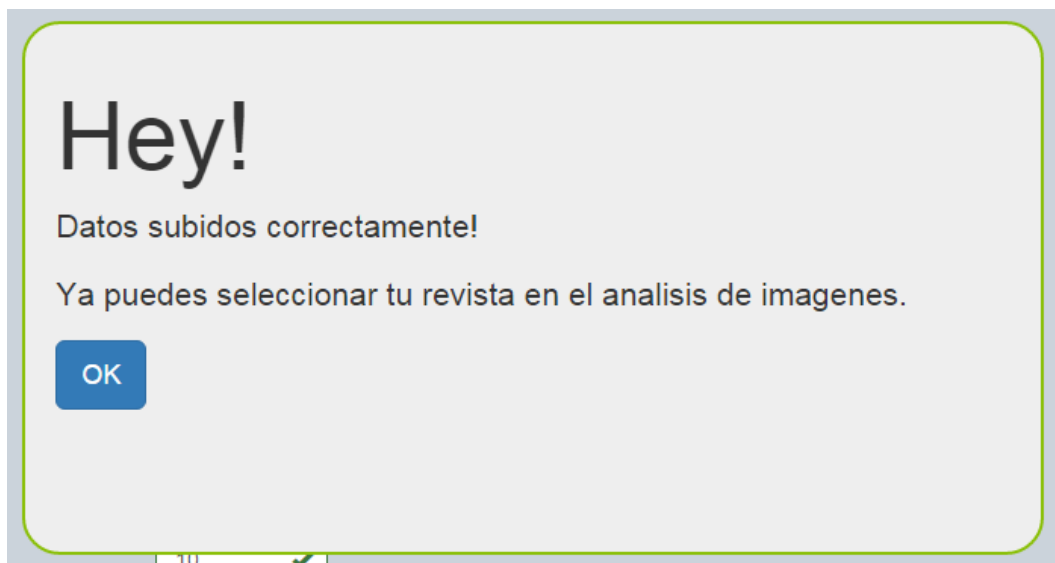


Figura 41. Mensaje de confirmación al añadir una nueva revista.

9.3 Anexo III: Casos de uso textuales

9.3.1 Analizar revista

Actores: Usuario autor.

Descripción: Un usuario autor quiere realizar el análisis de una imagen.

Pre-condición: -

Flujo básico:

1. El usuario selecciona su imagen y se realiza el análisis.
2. Se muestran los resultados al usuario.

Flujo alternativo:

1. Si no se ha seleccionado una imagen se muestra un mensaje de error.

Post-condición: -

9.3.2 Generar texto alternativo

Actores: Usuario autor.

Descripción: Un usuario autor quiere crear un texto alternativo para su imagen.

Pre-condición: Haber realizado el análisis de la imagen.

Flujo básico:

1. El usuario responde a las preguntas del formulario.
2. El usuario rellena la información necesaria para su texto alternativo.
3. El usuario descarga un pdf con la información que ha añadido.

Flujo alternativo:

2. En caso de no ser necesario un texto alternativo se le indica al usuario.

Post-condición: -

9.3.3 Añadir revista

Actores: Usuario editor.

Descripción: Un usuario editor quiere añadir su revista a la aplicación.

Pre-condición: -

Flujo básico:

1. El usuario introduce los datos en el formulario.
2. Si los datos son correctos se añade la revista a la aplicación.

Flujo alternativo:

2. Si los datos son incorrectos se mostrara el mensaje de error y el usuario podrá modificar los datos.

Post-condición: La nueva revista se añade a la aplicación.

9.4 Anexo IV: Presentación del test de usuario

Soy Manuel Chamorro, estudiante del Grado en Ingeniería informática que está realizando el TFG sobre la accesibilidad de las imágenes.

Has sido escogido para realizar un test en nuestra aplicación web para comprobar la accesibilidad de las imágenes.

El proyecto se encuentra en una fase de prototipo y aun no se encuentra en el mercado, por lo que cualquier cosa que realices tendrá ningún efecto real.

Realizaras dos tareas, primero una, y una vez finalizada pasaríamos a la siguiente. En la primera simularas ser el autor de un artículo que va a ser publicado en una revista y en la segunda serás el editor.

- **Tarea usuario autor:** Eres el autor de un nuevo artículo para la revista RevistaTfg. En este artículo que estás redactando quieres incluir una imagen para ilustrar el contenido. Para poder publicar esta imagen en la revista antes debes comprobar que cumple los requisitos mínimos para la revista. Para poder comprobarlo con facilidad y rapidez has decidido utilizar nuestra aplicación web. Lo primero que deberás hacer es acceder a la página web donde deberás seleccionar tu imagen, la revista y proceder a realizar su análisis. Deberás comprobar que la imagen cumple todos los requisitos mínimos, si es así, procederás a la creación del texto alternativo para la imagen. Una vez finalices el texto alternativo tendrás que descargarlo en tu ordenador para finalizar la tarea.
- **Tarea usuario editor:** Eres el editor de una revista, para poder cumplir con la legislación vigente necesitas que todas las imágenes que se publiquen en tu revista cumplan unos requisitos mínimos para ser consideradas accesibles. Para facilitar que los autores pueden comprobar con facilidad la accesibilidad de sus imágenes has decidido añadir la información de tu revista a nuestra página web.

Para realizar esta tarea deberás acceder a la página web y rellenar el formulario con la información requerida, una vez finalizado deberás subir esta información para que los autores puedan acceder a ella.

La información que debes añadir es la siguiente:

- Nombre: tu nombre
- Correo: revista@gmail.com
- Nombre revista: inventado
- Formatos aceptados: .png, .bmp
- Tipos de imagen y DPI(puntos por pulgada): color->72
- Resolución: 1024x 720 pixeles
- Tamaño: 12x12 inches(pulgadas)
- Peso: 2000 KB
- Número palabras en leyenda: 10 palabras
- Número de palabras en el título: 50 palabras

Si no tienes ninguna duda pasaremos a entregarte esta hoja de consentimiento para poder realizar el test.

9.5 Anexo V: Consentimientos para los test de usabilidad

Usuario 1

Por favor le esta hoja y fírmala.

En este test de usabilidad, le pediremos:

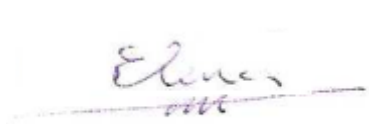
- Mantener una pequeña conversación con usted.
- Realizar una tarea en una página web.
- Rellenar un pequeño formulario de satisfacción

La participación en este estudio de usabilidad es voluntaria. Toda la información será estrictamente confidencial. Las descripciones y los resultados se podrán utilizar para ayudar a mejorar la página web. No obstante en ningún momento su nombre o cualquier tipo de identificación será utilizada.

Puede retirar su consentimiento y dejar de participar en cualquier momento.

He leído y entendido la información en este formulario y se han respondido a todas mis preguntas.

Firma del usuario



Firma del diseñador



Fecha

15/1/16

Usuario 2

Por favor le esta hoja y fírmala.

En este test de usabilidad, le pediremos:

- Mantener una pequeña conversación con usted.
- Realizar una tarea en una página web.
- Rellenar un pequeño formulario de satisfacción

La participación en este estudio de usabilidad es voluntaria. Toda la información será estrictamente confidencial. Las descripciones y los resultados se podrán utilizar para ayudar a mejorar la página web. No obstante en ningún momento su nombre o cualquier tipo de identificación será utilizada.

Puede retirar su consentimiento y dejar de participar en cualquier momento.

He leído y entendido la información en este formulario y se han respondido a todas mis preguntas.

Firma del usuario



Firma del diseñador



Fecha

15/11/16

Usuario 3

Por favor le esta hoja y fírmala.

En este test de usabilidad, le pediremos:

- Mantener una pequeña conversación con usted.
- Realizar una tarea en una página web.
- Rellenar un pequeño formulario de satisfacción

La participación en este estudio de usabilidad es voluntaria. Toda la información será estrictamente confidencial. Las descripciones y los resultados se podrán utilizar para ayudar a mejorar la página web. No obstante en ningún momento su nombre o cualquier tipo de identificación será utilizada.

Puede retirar su consentimiento y dejar de participar en cualquier momento.

He leído y entendido la información en este formulario y se han respondido a todas mis preguntas.

Firma del usuario



Firma del diseñador



Fecha

16/7/16

Usuario 4

Por favor le esta hoja y fírmala.

En este test de usabilidad, le pediremos:

- Mantener una pequeña conversación con usted.
- Realizar una tarea en una página web.
- Rellenar un pequeño formulario de satisfacción

La participación en este estudio de usabilidad es voluntaria. Toda la información será estrictamente confidencial. Las descripciones y los resultados se podrán utilizar para ayudar a mejorar la página web. No obstante en ningún momento su nombre o cualquier tipo de identificación será utilizada.

Puede retirar su consentimiento y dejar de participar en cualquier momento.

He leído y entendido la información en este formulario y se han respondido a todas mis preguntas.

Firma del usuario



Firma del diseñador



Fecha

16/2/16

Usuario 5

Por favor le esta hoja y fírmala.

En este test de usabilidad, le pediremos:

- Mantener una pequeña conversación con usted.
- Realizar una tarea en una página web.
- Rellenar un pequeño formulario de satisfacción

La participación en este estudio de usabilidad es voluntaria. Toda la información será estrictamente confidencial. Las descripciones y los resultados se podrán utilizar para ayudar a mejorar la página web. No obstante en ningún momento su nombre o cualquier tipo de identificación será utilizada.

Puede retirar su consentimiento y dejar de participar en cualquier momento.

He leído y entendido la información en este formulario y se han respondido a todas mis preguntas.

Firma del usuario



Firma del diseñador



Fecha

17/1/18