



TRABAJO FIN DE GRADO
GRADO EN INGENIERIA INFORMATICA

MemeHub

Edición Online de Imágenes

Autor

Víctor González Argudo

Director

Juan Julián Merelo Guervós



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, Junio de 2021

MemeHub

Edición Online de Imágenes

Víctor González Argudo

Palabras clave: *Javascript, React.js, JSX, Editor de Imágenes, Software Libre*

Resumen

MemeHub Online Image Editing

Víctor González Argudo

Keywords: *Javascript, React.js, JSX, Image Editor, Open Source*

Abstract

D. **Tutora/e(s)**, Profesor(a) del ...

Informo:

Que el presente trabajo, titulado **Chief**, ha sido realizado bajo mi supervisión por **Estudiante**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a Junio de 2018.

El/la director(a)/es:

(nombre completo tutor/a/es)

Agradecimientos

Índice general

1. Introducción	15
2. Descripción del problema	17
3. Estado del arte	19
4. Análisis del problema	21
5. Planificación	23
5.1. Metodología utilizada	23
5.2. Temporización	23
5.3. Seguimiento del desarrollo	23
6. Implementación	25
7. Conclusiones y trabajos futuros	27

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Este proyecto es software libre, y está liberado con la licencia [\[4\]](#).
//los memes blablabla y editores en la actualidad blablabla
Por qué hay necesidad de mi aplicación

Capítulo 2

Descripción del problema

Este proyecto pretende facilitar en la medida de lo posible la edición de imágenes que requieren de una edición rápida y ligera, como es principalmente el caso de los memes.

Actualmente, la mayoría de usuarios utilizan aplicaciones locales, instaladas en el propio ordenador, grandes y pesadas para la edición de imágenes que requieren de cambios simples, como puede ser: añadir texto, superponer imágenes encima de una base o crear formas geométricas para resaltar alguna parte de la imagen base.

Los programas de edición locales, como Photoshop o MSPaint, son pesados, con largos tiempos de carga y lo peor, requieren descargar tanto las fuentes como almacenar el resultado de forma local.

El objetivo de este proyecto es dar una solución mejor a este problema, siguiendo la tendencia actual del desarrollo en la actualidad, que son las aplicaciones web (SaaS, Software as a Service), evitando tener que efectuar ninguna descarga y dando la posibilidad de cargar, editar y exportar sin necesidad de emplear el almacenamiento local.

Capítulo 3

Estado del arte

La motivación de este proyecto surge precisamente por el estado actual de los editores de imágenes. La mayoría de usuarios, edita el contenido que sube online desde una aplicación local instalada en su ordenador (Microsoft Paint, Photoshop), o en el caso de los que emplean aplicaciones en la nube, descargan sus imágenes para luego resubirlas a sus redes. Concretamente los memes suelen ser ediciones especialmente simples, donde se añade un texto, se superponen varias imágenes o se dibuja algo encima de una imagen base.

Este proyecto tiene como objetivo tratar este problema y evitar en la medida de lo posible el uso del almacenamiento local de imágenes, tanto para crear, como para subir el contenido ya editado.

Capítulo 4

Análisis del problema

//Estudio de viabilidad (tiene sentido) //Perfiles implicados, trabajadores, usuarios, //que va a hacer el cliente (tendra exito si tiene una experiencia fluida)
//Casos de uso //Req fun //Req no fun

Capítulo 5

Planificación

5.1. Metodología utilizada

Para la realización de este proyecto se ha seguido una metodología ágil o 'Agile', esta metodología es especialmente buena para los proyectos que necesitan de mucha flexibilidad, por ende, por la naturaleza de algo como una aplicación de edición de imagen, que es algo que invita fácilmente a añadir nuevas funcionalidades y a cambiar o adaptar antiguas al nuevo contenido que se añada, ha sido escogida para el desarrollo de este trabajo.

Para la metodología ágil escogida

5.2. Temporización

5.3. Seguimiento del desarrollo

//kanban

Capítulo 6

Implementación

Para comenzar con la implementación, se partió del código elaborado como demo en la fase de planificación. Está bastante extendido en el desarrollo de React emplear el paquete create-react-app [2].

Es un paquete creado por Facebook que se instala con el gestor de paquetes de node (npm) que crea una aplicación de React vacía con todo lo básico necesario más algunos scripts que ayudan en el desarrollo y en el despliegue, como por ejemplo, el script react-run, para ejecutar el servidor de desarrollo, que se recompila automáticamente al detectar cambios y hace el desarrollo mucho más fluido o el build para crear una versión release optimizada.

Por si sólo, React es una biblioteca para construir interfaces de usuario que cuenta con módulos extra que permiten extender su funcionalidad, es decir, no es un framework, como otras tecnologías empleadas para el diseño de interfaces, como Vue.js o Angular.js, sin embargo con estos módulos podemos ampliar las funcionalidades de la biblioteca React dependiendo del uso que vayamos a darle, en nuestro caso como queremos crear una aplicación web, necesitamos interactuar con el 'Document Object Model' (DOM [1]), que es la estructura de documentos HTML que genera React y procesa el navegador, por ende y tras la creación de la plantilla inicial en React, instalamos los módulos de React que permiten interactuar con él, react-dom [3].

También necesitamos un módulo que permita a React cambiar la página generada dependiendo de la ruta a la que accede el usuario, en el caso de nuestra aplicación, siempre queremos mostrar la página del editor así que debemos redirigir todas las rutas al componente principal <Editor/> que veremos en profundidad más adelante (aunque se planea añadir más páginas en un futuro). Para ello, empleamos el módulo 'react-router-dom' [5] que es el módulo más utilizado en la comunidad de React para implementar esta funcionalidad.

En un primer lugar se pensó introducir una página de error, pero ya que en principio solo hay una página en toda la web, se han redirigido todas las rutas a la misma, el componente principal.

```
1  <Router>
2    <Switch>
3      <Route exact path="/" component={Editor} />
4      <Route exact path="/editor" component={Editor} />
5      <Route component={Editor} />
6    </Switch>
7  </Router>
```

Aunque bastaría con emplear una única ruta se han especificado también las rutas '/' y '/editor' ya que, como se ha comentado antes, se planea extender la cantidad de páginas en un futuro.

//konva

Capítulo 7

Conclusiones y trabajos futuros

Si estoy satisfecho
Has conseguido cumplir los objetivos
se ha seguido la planificación

Bibliografía

- [1] Document object model. https://developer.mozilla.org/es/docs/Web/API/Document_Object_Model/Introduction.
- [2] Facebook. Create react apps with no build configuration. <https://github.com/facebook/create-react-app>.
- [3] Facebook. React package that implements dom specific methods. <https://es.reactjs.org/docs/react-dom.html>.
- [4] Free Software Foundation. GNU General Public License. <http://www.gnu.org/licenses/gpl.html>.
- [5] ReactTraining. React package for routing react web apps. <https://reactrouter.com/web/guides/quick-start>.