

MEMORIA ESCRITA DEL PROYECTO

CFGS Desarrollo de Aplicaciones Web

DgToolbox

Autor: Gabriel Cruz Mayo

Tutor: Tamara González Gómez

Fecha de entrega: 01/12/23

Convocatoria: 1s2324

Documentos del proyecto: https://drive.google.com/drive/folders/1B5hIUUyI0Pq-fJWZtdmhKvRZ6z5ao-z1?usp=drive_link



Índice de contenidos

1. Introducción	3
1.1. Motivación.....	3
1.2. Abstract	3
1.3. Objetivos propuestos (generales y específicos)	4
2. Estado del Arte	5
3. Metodología usada	9
4. Tecnologías y herramientas utilizadas en el proyecto	10
5. Planificación, Diagnóstico y Contexto Laboral.....	13
6. Análisis del proyecto	14
7. Diseño del proyecto	29
8. Despliegue y pruebas	40
9. Conclusiones	42
10. Vías futuras.....	43
11. Bibliografía/Webgrafía	44
12. Anexos	47
12.1. Glosario.....	47
12.2. Diagrama de Casos de uso	48
12.3. Wireframes.....	49
12.4. Manual de usuario.....	51
12.5. Manual de despliegue	57

1. Introducción

1.1. Motivación

La motivación para llevar a cabo este proyecto en concreto es la inspiración de plataformas como google drive, aplicaciones que son de vital importancia en mi día a día, y me motiva la idea de poder desarrollar una app que sea capaz de recibir no solo datos de texto plano, sino imágenes y persistirlas en un servidor dándole la mayor de la seguridad posible dentro de mis conocimientos.

Otro aliciente es el reto técnico que se me presentaba tratar de llevar a cabo un concepto parecido a estas aplicaciones, usando los conocimientos adquiridos durante el CFGS ¹ DAW ², además mi orientación profesional va encaminada al backend y el desarrollo de la lógica de negocio de aplicaciones web, persistencia de datos, y este proyecto me parecía contener esos aspectos en común.

Este proyecto me permitiría no solo afianzar conocimientos de todo el curso sino, aprender una nueva manera de persistir archivos con un nivel de opacidad y privacidad alto debido a una capa de encriptación y mejorar mi comprensión sobre persistencia, comunicación cliente-servidor y manipulación de datos y archivos.

1.2. Abstract

DgToolbox is a web application that can host in its interior a variety of embedded, modular and independent applications to give solution to different office needs of the user, in this project a complex word processor and an image gallery are developed, which can be used to save texts and images in the server achieving a remote storage and also adds the ability to download these files. Allowing the user to access from any computer to your account and thus to your files without having to worry about storage capacity or transport your files physically. Throughout the life of the application an infinite amount of applications could be added to provide solutions to different user needs and for any computer without taking up hard disk space and regardless of the technical characteristics of the client computer (www.DeepL.com).

¹ Ciclo formativo de grado superior

² Desarrollo de aplicaciones web

1.3. Objetivos propuestos (generales y específicos)

Objetivos Generales

Se desarrollará una aplicación web con una estructura modular en donde la aplicación principal se comportara como un recipiente de otras sub-aplicaciones dedicadas a ofrecer herramientas ofimáticas y multimedia al usuario en su día a día, bajo un perfil de usuario que le permita adaptar su experiencia de uso y persistir archivos personales en el servidor, además posteriormente podrá descargar dichos archivos desde el dispositivo desde el cual se conecte.

En una primera versión la aplicación contara con un procesador de texto y una galería de imágenes. Además contará con una sección para el rol de administrador de la aplicación.

Objetivos específicos

- Guardar registros de usuario en BBDD para la verificación y creación de sesiones de uso en la aplicación.
- Implementación de las funciones básicas por parte del usuario que podemos encontrar en cualquier aplicación web de hoy en día: posibilidad de iniciar sesión, cerrar sesión actual y borrar la cuenta en uso por el propio usuario.
- Permitir el acceso a la plataforma con un perfil de invitado y permitir el uso limitado de la plataforma restringiendo la posibilidad de persistir datos en el servidor.
- Encriptar las contraseñas en la BBDD y poder recuperarlas correctamente para la autenticación y autorización de usuario.
- Encriptación de los archivos de imágenes y texto para otorgar una mayor privacidad a los archivos de los clientes.
- Generar un procesador de texto funcional y lo más completo posible. Con la posibilidad de persistir, borrar y recuperar textos en una BBDD alojada en el servidor.
- Desarrollar una galería de imágenes con la capacidad de guardar, borrar y recuperar imágenes en distintos formatos en una BBDD alojada en el servidor.
- Implementar que todos los procesos de persistencia, borrado, modificación y recuperación de las BBDD se realicen de manera asíncrona mediante AJAX.

- Generar un área de administrador desde la que pueda otorgar y quitar roles de administrador entre las cuentas de la BBDD, borrar usuarios así como ver el listado de usuarios registrados.
- Realizar una interfaz gráfica con Bootstrap lo más amena y moderna posible haciendo uso de iconos y espacios en blanco para obtener la mayor eficiencia y conseguir que sea lo más intuitiva posible.
- Desarrollar la posibilidad el usuario pueda descargar en su sistema los archivos persistidos en la BBDD (textos e imágenes).

2. Estado del Arte

DgToolbox se trata tanto de una aplicación web de ofimática como de un servicio de persistencia de datos en un servidor remoto a lo que se le denomina, aunque este ejemplo en algunos aspectos pueda llegar a ser algo primitivo ya que es una primera versión, como guardado en la nube.

Pero primero habría que explicar que es persistir datos en la nube. El almacenamiento en la nube es un servicio que le permite almacenar datos enviándolos a través de Internet u otra red a un sistema de almacenamiento externo mantenido por un tercero. Hay cientos de sistemas de almacenamiento en la nube diferentes, desde el almacenamiento personal, que almacena o realiza copias de seguridad de los correos electrónicos, fotos, vídeos y otros archivos personales de los usuarios, hasta el almacenamiento empresarial, que permite a las empresas utilizar aplicaciones de almacenamiento en la nube. Las soluciones de respaldo permiten a su empresa transferir, almacenar o compartir archivos de datos de forma segura entre oficinas (Microsoft, 2023).

En el mercado existen a juicio del investigador 3 grandes empresas y sus consiguientes plataformas que lideran el mercado de guardado en la nube. La más antigua OneDrive (2007), posteriormente Dropbox (2008) seguida de y Google Drive (2012) como la última en aparecer (Dropbox, 2023; News Center Microsoft Latinoamérica, 2022; Wikipedia, 2023).

De estas solo ofrecen soluciones ofimáticas integradas o conectadas en la aplicación de persistencia de datos, Google Drive con Google Workspace y OneDrive con Microsoft365, quedando finalmente Dropbox fuera en este aspecto.

Si hablamos de las soluciones ofimáticas de GoogleDrive nos encontramos un modelo totalmente integrado en su plataforma, gratuito, aunque no completo de algunas de las aplicaciones de Google Workspace, como son su procesador de texto, hoja de cálculo, presentaciones, aplicación para dibujar, editor de formularios entre otras. Además incluye

un Marketplace donde podemos conectar Google Drive a otras aplicaciones de terceros que aumentan la cantidad de servicios que puedes conectar a Google Drive.

En cuanto a One Drive de Microsoft ofrece el antiguo paquete de office con todas sus aplicaciones de manera gratuita conectándose al servicio de Microsoft365, pero en esta versión gratuita muchas de sus funciones se encuentran limitadas, y se ha de acceder a los planes de pago de Microsoft365 para desbloquearlos.

Comparando estos dos servicios en su aspecto de soluciones ofimáticas que ofrecen, encontramos que mientras google apuesta por un uso gratuito y completo de algunos de los servicios de Google Workspace, y añade un marketplace que permite que amplíe su oferta de soluciones dentro de su servicio de guardado en la nube de manera ilimitada, haciendo muy interesante el que terceros traten de adecuar sus aplicaciones para que se puedan conectar al funcionamiento de esta plataforma causando una retroalimentación muy beneficiosa tanto para google ,como para los terceros desarrolladores y usuarios.

Sin embrago aunque Microsoft en ese aspecto no tenga una dinámica tan competitiva aporta un punto importante y es que su software ofimático es un estándar a día de hoy y en mi experiencia propia las aplicaciones de google carecen de la profundidad que aportan las de Microsoft en sus versiones de pago.

Google entrega más variedad y adaptabilidad, mientras que en este aspecto Microsoft aporta un menor número de posibilidades, a cambio de aplicaciones ampliamente probadas y depuradas con una profundidad en su funcionamiento mucho mayor, más adecuadas para un uso profesional.

Respecto al almacenamiento remoto todas estas plataformas ofrecen un uso gratuito y limitado a una determinada capacidad de almacenamiento. A este respecto, Dropbox es la plataforma que menos ofrece de manera gratuita siendo 2 GB³, One Drive sube a 5 GB este ofrecimiento, mientras que Google Drive es la que más cantidad de almacenamiento gratuito ofrece con 15 GB. Las tarifas de pago serían las siguientes:

-Google Drive:

100 GB: 1,99 euros/mes ó 19,99 euros/año, 200 GB: 2,99 euros/mes ó 29,99 euros/año.

2 TB: 9,99 euros/mes ó 99,99 euros/año, 5 TB: 24,99 euros/mes ó 249,99 euros/año.

10 TB: 49,99 euros/mes, 20 TB: 99,99 euros/mes, 30 TB: 149,99 euros/mes.

One Drive:

100 GB: 2 euros/mes ó 20 euros/año, 1 TB: 7 euros/mes ó 69 euros/año.

³ es una unidad de almacenamiento de información estandarizada equivale a 10⁹ bytes

6 TB: 10 euros/mes ó 99 euros/año.

Dropbox:

2 TB: 1.99 euros/mes ó 119,88 euros/año ,3 TB: 19,99 euros/mes ó 198,96 euros/año.

Precios por usuario a partir de 3 usuarios en un mismo plan:

5 TB: 14,50 euros/mes ó 144 euros/año, 15 TB: 21,50euros/mes ó 216 euros/año.

El último apartado, y quizás el de mayor importancia a la hora de elegir donde guardar información privada, es el de la seguridad de los datos.

Por un lado en Dropbox los datos en tránsito se cifran con SSL y mediante AES de 128 bits. Cuando los archivos están en reposo, Dropbox usa AES de 256 bits para el cifrado. En caso de robo o pérdida de un dispositivo se puede desvincular fácilmente y así mitigar el riesgo. Dropbox no pueden ver el contenido de los archivos, aunque sí acceder a los metadatos para dar soporte técnico si fuera necesario. También ofrece verificación en dos pasos.

Google Drive cifra los datos encriptados con SSL cuando se transfieren, y usa AES 256 bits tanto en tránsito como en reposo, en este sentido hay diferencias respecto a Dropbox. También ofrece como es común hoy día verificación en dos pasos.

Finalmente, OneDrive, podemos encontrar diferencias con los anteriores, para empezar, no cuenta con cifrado en reposo para la versión gratuita. Sí ofrece cifrado en tránsito mediante SSL. No obstante, en versiones de pago cuenta con cifrado en reposo AES 256 bits individual para cada archivo. Esto significa que si se compromete uno de ellos, no afecta a todos.

Si tuviéramos en cuenta únicamente la seguridad, podríamos decir que no existen realmente diferencias para el usuario medio, exceptuando la versión gratuita de One Drive que es claramente más insegura. Una vez nos metemos en planes de pago las diferencias son mucho más atenuadas, teniendo quizás un mayor valor el que aporta DropBox que tiene un servicio técnico más completo y una encriptación más pesada pero algo más segura. Sin tener en cuenta las políticas en cuanto a planes de empresa de cada una de ellas, que quizás podría variar este aspecto según los tratos a los que se lleguen particularmente (Dropbox, 2023; Microsoft, 2023; Google Drive, 2023).

Como podemos ver al comparar los planes de suscripción y teniendo en cuenta que nuestra aplicación se enfocara principalmente y en un inicio al usuario particular. Google

Drive es la opción más competitiva de las que hay en cuanto a variedad de opciones en la suscripción y precios.

Además tiene una característica que considero con un potencial muy importante para diferenciarse de la competencia en este target de cliente y es un carácter más cercano a la comunidad y abierto tanto en su uso de las aplicaciones base como por su marketplace.

La baza más fuerte de su competencia más directa, que sería OneDrive, es que el software ofimático es superior a día de hoy.

Entonces, ya hemos señalado quienes son los líderes en este mercado y hemos analizado sus puntos fuertes y sus puntos débiles. Y llego a la conclusión de que un software que quiera irrumpir en el panorama actual deberá enfocarse en los siguientes puntos.

Primero deberá estar a la altura en cuanto los precios de mercado en cuanto a costo por GB, manteniéndose pegado a Google Drive y su estilo de segmentar un número considerable de planes para flexibilizar su acceso. Segundo ofrecer un software ofimático por lo menos a la altura entre la profundidad de Microsoft y el presentado por google, manteniendo su carácter gratuito, a la vez que se incentiva a su comunidad de seguir aportando utilidades a través de un marketplace interno, facilitando API's para su incorporación al entorno que estemos creando en común, creando un sistema de comisiones que reparta beneficios sobre las aplicaciones mejor valoradas y más descargadas.

Y por último y tercer punto, garantizar la seguridad de los datos de la manera más minuciosa posible y destacar en ese aspecto o por lo menos estar entre los estándares más altos. Con igualar los estándares sería beneficioso ya que las dos líderes son grandes empresas que en algún momento han suscitado dudas y temores por parte de los usuarios por la ingente cantidad de datos personales que llegan a manejar entre todas sus plataformas, a pesar de las políticas públicas que tienen sobre la seguridad y la restricción de acceso por parte de ellos a los datos de los usuarios. Como recientemente ocurre con el caso del lanzamiento de la IA Microsoft Copilot, señalada por poder estar usando código de repositorios privados en Github y otros datos de naturaleza privada.

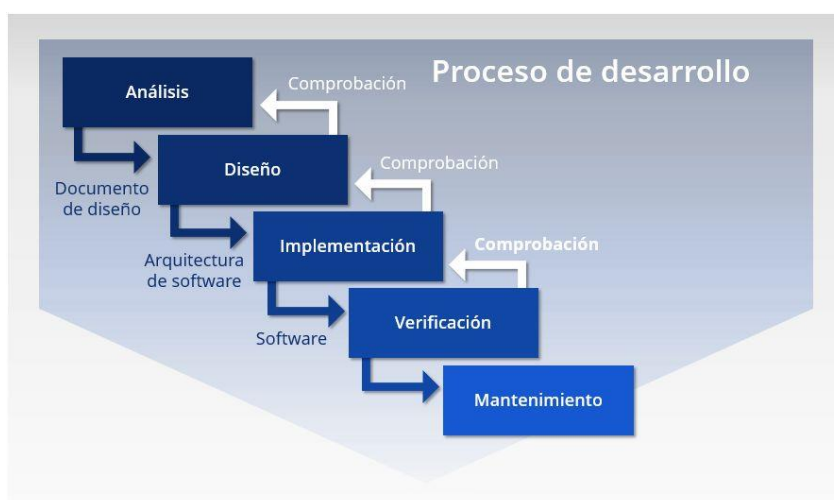
Concluyo con la opinión que una compañía que cumpla con lo anterior y además demuestre que son una empresa dedicada por y para este servicio, como ocurre con Dropbox (cuyos precios se hayan más alejados del estándar y no tiene soluciones ofimáticas ni aplicaciones internas), conseguiría colocarse en la cima de este tipo de servicios por su variedad de oferta, seguridad y confianza.

3. Metodología usada

La metodología utilizada para la realización de este proyecto fue el método de desarrollo en cascada con retroalimentación.

En este modelo es una variación del modelo en cascada tradicional en el que las etapas para el desarrollo de software tienen un orden establecido: Análisis, Diseño, Codificación, pruebas y Mantenimiento. Además para empezar una etapa, es necesario finalizar la etapa anterior. Al concluir cada etapa se realiza una revisión para comprobar si se puede pasar a la siguiente

En el modelo en cascada con realimentación, se puede producir una realimentación entre etapas. En el caso que en alguna de las etapas se detectaran fallos (cambios en los



requisitos, errores no contemplados en fases

anteriores, etcétera), en ese caso, podremos retornar a la etapa anterior para realizar los ajustes pertinentes. A esto se le conoce como realimentación, pudiendo volver de una etapa a la anterior o a varias etapas anteriores según se vaya viendo necesario al reajustar (Ilerna Online S.L, 2021).

A lo largo del desarrollo podemos distinguir:

-Análisis: Se analizara las características operativas del software y marcaremos las restricciones y viabilidad del proyecto o sección de este a desarrollar en el ciclo en el cual nos encontremos (Ilerna Online.S.L, 2021).

En este punto desarrollaremos los diagramas de clase, diagramas E-R⁴, casos de uso, Requisitos funcionales y no funcionales de cada fase del proyecto.

-Diseño: Tendremos que componer la forma para solucionar el problema. Traduciremos los requisitos funcionales y los no funcionales del análisis en una representación de software, utilizando los distintos diagramas para establecer una guía de ejecución en la práctica.

⁴ Diagrama de entidad – relación, más información en el glosario

Los diagramas que usaremos en esta fase serán diagramas de flujo, cajas de decisión, pseudocódigo y diagramas de cajas. Además, se añadirán esquemas del diseño de la UI⁵, así como análisis en detalle de los componentes más importantes del proyecto. (Ilerna Online.S.L, 2021).

-Codificación: En esta fase nos encargaremos de utilizar los datos del diseño y transformarlo en lenguaje de programación. A estas instrucciones las llamaremos código fuente. En este apartado también podemos encontrar además de la codificación pura de código fuente aspectos como: Marcar pautas de buenas prácticas y código limpio, organización de la estructura de ficheros y normas de estilo (Ilerna Online.S.L, 2021).

-Pruebas: En esta fase del ciclo, trataremos de encontrar errores en el código, en las especificaciones o en el diseño. Durante esta fase, se realizan las siguientes tareas:

-Verificación: Comprobaremos que el software está cumpliendo con los requerimientos, debe haber una prueba por cada uno de ellos.

Por ejemplo, que las imágenes se persisten en la BBDD⁶ cuando pulsamos el botón de guardar o cuando creamos un nuevo usuario y este queda persistido.

-Validación: Buscaremos alguna situación en la que el software no cumpla con las especificaciones o sea incorrecto (Ilerna Online.S.L, 2021).

Por ejemplo, buscaremos maneras en las que el sistema de login falle, como que no distinga entre mayúsculas o minúsculas, que el sistema no capture excepciones como tratar de borrar o buscar algo que no existe en BBDD (imágenes, textos o usuarios).

4. Tecnologías y herramientas utilizadas en el proyecto

JavaScript

Javascript es un lenguaje de programación interpretado, un dialecto derivado del estándar ECMAScript⁷. Se caracteriza como orientado a objetos, basado en prototipos, imperativo, de tipado débil y dinámico. Todos los navegadores modernos leen el código JavaScript insertado en las páginas web. Para actuar sobre la página web se proporciona al lenguaje JavaScript una implementación del Document Object Model (DOM⁸). Javascript es el único lenguaje de programación comprendido de manera nativa por los navegadores. (wikipedia, 2023).

⁵ Acrónimo en inglés de interfaz gráfica

⁶ Acrónimo de bases de datos

⁷ Especificación sobre estándares de lenguajes de programación. Más información en el glosario

⁸ Permite acceso para acceder, añadir y cambiar dinámicamente contenido estructurado en documentos, Más información en el glosario

Elegimos este lenguaje para dar funcionabilidad dinámica a nuestra web y comunicarnos de manera asíncrona con el servidor mediante la tecnología AJAX que proporciona JQuery.

JQuery

“jQuery es una pequeña y rápida biblioteca de JavaScript con muchas funciones. Permite que aspectos como la navegación y manipulación de documentos HTML, el manejo de eventos, la animación y la tecnología AJAX sea mucho más sencillo con una API fácil de usar esta librería que funciona en una gran cantidad de navegadores”. (OpenJS Foundation, 2023).

Esta librería se eligió para facilitar el uso de JavaScript y AJAX consiguiendo tener un código más limpio y legible.

AJAX

AJAX, es una tecnología que permite realizar peticiones al servidor desde JavaScript de manera asíncrona, es decir, realizará la petición y la petición se desarrollara en segundo plano, y así el usuario podrá seguir navegando por la web o aplicación mientras se entrega la respuesta recibéndola a demás sin tener que recargar la página.

Las tecnologías que utiliza por debajo para realizar esto son: XHTML, CSS, DOM, XML, XSLT, JSON, XMLHttpRequest, JavaScript (Ilerna Online S.L, 2021)

CSS

CSS (Cascading Style Sheets), es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado y escrito en un lenguaje de marcado.

Está diseñado principalmente para mantener separadas del contenido del documento, características tales como las capas, los colores, las fuentes, transiciones y animaciones además de un largo etc.

Esta separación permitirá modularizar más nuestra aplicación y disminuir la repetición de código ya que va a permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo “.css”. (wikipedia, 2023)

PHP

PHP (Acrónimo recursivo de Hypertext Preprocessor), es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser incrustado en páginas HTML. El objetivo principal de este lenguaje es permitir a los desarrolladores web escribir dinámica y rápidamente páginas web generadas; aunque se puede hacer mucho más con PHP. (PHP, 2023)

Se eligió en este proyecto para realizar la programación del lado del servidor, lógica de negocio y persistencia de datos en BBDD⁹.

HTML

Lenguaje de marcado que es interpretado por navegadores web para mostrar visualmente los resultados al usuario. En específico usaremos la versión HTML5 que trae las últimas novedades respecto a programación web y el uso de etiquetas que necesitaremos al usar Bootstrap.

Bootstrap

Es un framework¹⁰ de código abierto que permite desarrollar interfaces gráficas con diseño responsive, permitiendo que se visualicen exitosamente en cualquier dispositivo y cumple además con la función de facilitar enormemente la construcción de estas interfaces y su estructuración mediante un sistema de cuadrícula estructurado en 12 columnas, además de variables Sass¹¹, componentes preconstruidos y funciones JavaScript (Bootstrap).

FontAwesome

Se trata de un framework que fue pensado para su uso junto con Bootstrap. Este permite el acceso a iconos y fuentes gratuitos y de pago, los cuales puedes modificar enteramente usando CSS y darles animaciones, con el uso de clases en sus etiquetas como en Bootstrap. Utiliza CSS y Less¹² para llevar a cabo su funcionamiento (FontAwesome, 2023).

API googleFonts

Se trata de una API a la que conectarse a través de una hoja de estilo y de la que obtendré diferentes fuentes que proporciona google para su uso. En este proyecto únicamente se utiliza para la fuente usada como tipografía en la app “Roboto”.

Visual Studio code

Es un editor de texto, específico para codificar y desarrollar software, que permite la codificación de lenguaje PHP y JS de manera fácil y muy completa gracias a los plugins que puedes obtener en su marketplace, que hacen que se asemeje mucho a manejar un IDE en términos de escritura de código.

⁹ Acrónimo de base de datos.

¹⁰ Anglicismo que significa entorno de trabajo. Más información en glosario.

¹¹ Metalenguaje de hojas de estilo en cascada (CSS).

¹² Lenguaje de programación de hojas de estilo en cascada (CSS).

Git: Sistema de control de versiones, que permitirá trabajar con seguridad, pudiendo revertir cambios, proteger el código trabajando en ramas de pruebas antes de unir los cambios a la rama principal o volver a estados anteriores de desarrollo.

WAMP

Se trata de un conjunto de herramientas que permiten desplegar un servidor local en tu equipo usando Apache, y un servidor MySQL integrado que será el que use en este proyecto, justo a su gestor de bases de dato phpMyAdmin.

5. Planificación, Diagnóstico y Contexto Laboral

La planificación de este proyecto empezó desde el día que mande la propuesta, ya que por los plazos dados y al ver que no fue resuelta en las 24 horas siguientes, me pareció que las fases de análisis y diseño podrían comenzar desde ese momento y en el caso de ser aceptado habría ganado un valioso tiempo y en contrario podría hacer los arreglos pertinentes y modificar el trabajo hecho hasta el momento.

Realice un primer diagrama de Gantt para decidir los plazos y orden en el que empezaría a tratar las distintas partes del proyecto, el cual al ser teórico y teniendo en cuenta el modelo de desarrollo elegido, situarían las fases de análisis, diseño, codificación y pruebas una detrás de la otra en principio no empezarían si todo iba bien sin haber acabado la anterior.

La documentación podría empezar desde el primer día si la cantidad de horas disponibles eran favorables rellenando partes como la introducción, y las tecnologías mencionadas en la propuesta, en el diagrama aparece como una barra discontinua aunque ya preveía que no trabajaría todos los días en ella y que el grueso lo realizaría al ir acabando las pruebas, aunque la intención es ir avanzándolo poco a poco antes de este punto justo con el video de presentación en el cual cuento también el tiempo necesario para hacer las diapositivas

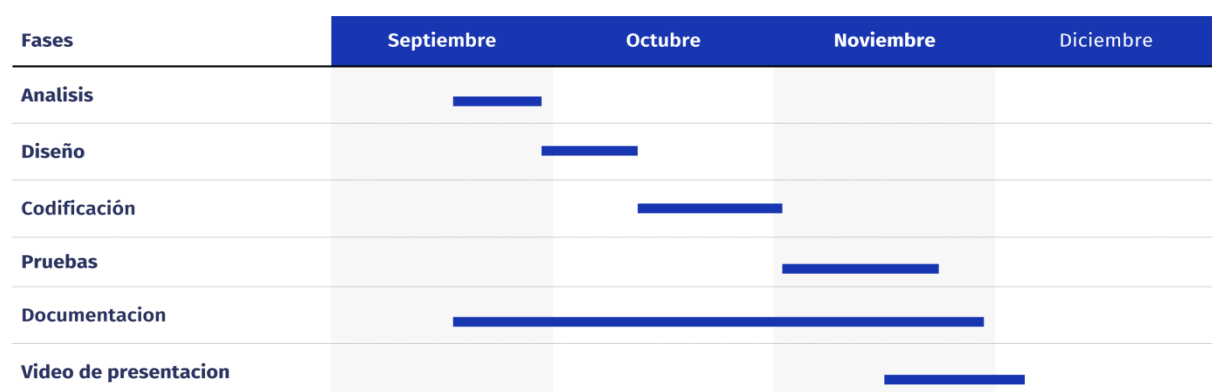


Imagen 2 Diagrama de Gantt inicial (Canva, 2023)

Finalmente y como era de esperar hubo variaciones en el diagrama de Gantt, sobre todo respecto a la etapa de pruebas, como se puede observar en el diagrama, en la cual tuve que regresar a la fase de diseño y modificar una sección que tenía que ver con la estructura de ficheros, añadiendo una carpeta extra que no estuvo prevista al inicio para un almacenamiento temporal de los recursos a descargar de la base de datos. Posteriormente volver a la fase de codificación y a lo largo de los últimos días hubo varias ocasiones que estuve saltando de las pruebas a la codificación por problemas a la hora de pasar las pruebas.

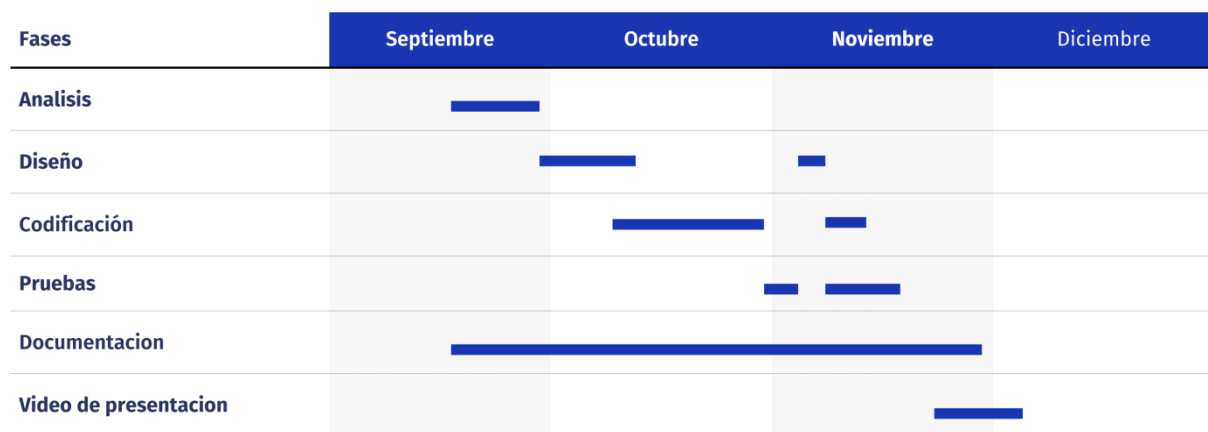


Imagen 3 Diagrama de Gantt final (Canva, 2023)

Contexto laboral:

Esta aplicación web, tiene como objetivo ser distribuida como un freemium como sus competidores mencionados en el apartado de estado del arte, incluso dejando disfrutar de herramientas de la aplicación sin darse de alta en nuestra aplicación de modo que podamos suscitar el interés de ir desbloqueando esas otras funcionalidades, como persistir sus datos en nuestro servidor entre otras que se agregarían más adelante, y ganando la confianza del cliente con esa transparencia de nuestros servicios.

Esta aplicación daría además trabajo a todo un equipo de desarrollo de software para llevar a cabo mantenimiento y evolutivos, un equipo de marketing, gestores y profesionales que permitan mantener y circunscribir todas nuestras actividades en el marco legal.

6. Análisis del proyecto

En esta sección analizaremos los detalles más generales del proyecto, que dirigirán en las fases posteriores el camino a seguir y los objetivos a cumplir para que el proyecto tenga las funcionalidades con las que fue concebido. En esta sección encontraremos los requisitos

funcionales, los diagramas de E-R, diagramas de caso de uso y tarjetas de casos de uso y el diagrama de clases.

Requisitos Funcionales:

RF 1: Inicio por defecto del cliente como usuario invitado.

RF 2: Acceso limitado del cliente como usuario invitado al procesador de texto de la aplicación de manera limitada permitiendo escritura y descarga.

RF 3: Permitir registro del cliente como usuario de la aplicación.

RF 4: Permitir login de un usuario previamente creado en la aplicación y permitir logout de un usuario con sesión activa.

RF 5: permitir al cliente borrar permanentemente al usuario de la base de datos y de la aplicación.

RF 6: Crear un procesador moderno y funcional que permita crear, editar y dar formato a textos.

RF 7: Implementar la función de guardado de textos en la base de datos, recuperación de textos desde la base de datos, borrado de textos y la posibilidad de ser descargados al equipo desde la base de datos.

RF 8: Implementar una galería de fotos persistidas en una base de datos.

RF 9: Implementar guardado de imágenes por parte de un usuario registrado en la base de datos, así como el borrado y descarga al equipo de imágenes persistidas.

RF 10: Persistir los textos e imágenes en el interior de la base de datos, no una referencia sino toda la información del objeto en la base de datos.

RF 11: Proteger las contraseñas de usuario, textos e imágenes con encriptación AES_128bits.

RF 12: Desarrollar un área de administrador solo accesible para los usuarios administradores.

RF 13: El área de administrador permitirá borrar usuarios y nombrar nuevos administradores.

RF 14: No se permitirá que un usuario administrador se borre a si mismo dentro del área admin.

RF 15: Una vez el usuario inicie sesión esta perdurara activa en todas las fases y estancias de la aplicación y recordara al usuario mientras no haga logout o no cierre el navegador, aunque cierre la pestaña.

RF 16: La Respuesta del servidor en el área admin así como en la galería y procesador de texto se ejecutara de manera asíncrona gracias a AJAX.

Requisitos no funcionales:

RNF 1: Asegurar la seguridad de los archivos persistidos en la base de datos así como de las contraseñas de usuario y mantene la integridad y coherencia de la aplicación.

RNF 2: Desarrollar una interfaz intuitiva y agradable para el usuario.

RNF 3: Mantener un estilo tipográfico y de diseño homogéneo entre las distintas herramientas.

RNF 4: Hacer uso de iconos siempre que sea posible y utilizar animaciones para hacer énfasis en la posición del cursor.

Diagrama Entidad-Relación:

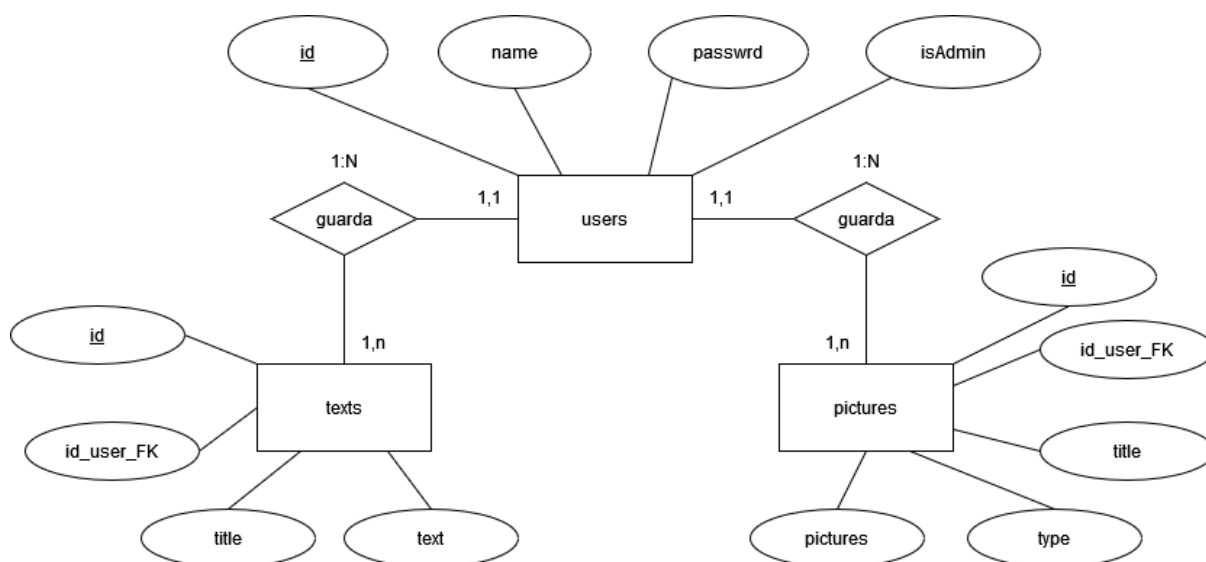


Imagen 4 Diagrama Entidad-Relación (Diagrams.net, 2023)

La base de datos de DgToolbox contara con tres tablas: Users, para los actores humanos del sistema, Texts para los textos persistidos y Pictures para las imágenes.

La estructura de estas de datos de las tablas es la siguiente:

Users:

id: Campo que hará de primary key, int de 11 dígitos de capacidad, con el atributo auto_increment.

name: Nombre de usuario, varchar¹³ de 20 caracteres de capacidad y con el atributo unique¹⁴.

passwd: Contraseña de usuario, transformada a un blob¹⁵ encriptado en AES 128bits.

isAdmin: indicará si el usuario tiene permisos de administrador, siendo 1 que es un admin y 0 que no lo es. Estructura de int de un dígito de capacidad y por defecto su valor será 0.

Todos los campos de la tabla users están creados con el atributo Not Null.

Texts:

id: Campo que hará de primary key, int de 11 dígitos de capacidad, con el atributo auto_increment.

id_user: Clave foránea que hace referencia al id de la tabla user y relaciona a los usuarios con sus textos. Estructura de int con 11 dígitos de capacidad.

title: Título del texto persistido en la base de datos, varchar de 120 caracteres de capacidad.

text: Aquí se almacenaran los textos del usuario, en el formato longblob y encriptado con AES 128bits.

Todos los campos de la tabla text están creados con el atributo Not Null¹⁶.

pictures:

id: Campo que hará de primary key, int de 11 dígitos de capacidad, con el atributo auto_increment.

id_user: Clave foránea que hace referencia al id de la tabla user y relaciona a los usuarios con sus imágenes. Estructura de int con 11 dígitos de capacidad.

title: Título de la imagen persistida en la base de datos, varchar de 120 caracteres de capacidad.

type: Indicara la extensión del tipo de archivo persistido (ejem: ".jpg"), tendrá la estructura de un varchar de 50 caracteres de capacidad.

picture: Aquí se almacenara las imágenes del usuario, en el formato longblob y encriptado con AES 128bits.

¹³ Formato constituido por una cadena de caracteres.

¹⁴ Indica que el campo debe de ser único en toda la tabla, no permitiría registros con el mismo valor

¹⁵ Formato que guarda la información en sistema binario.

¹⁶ No permite que en ese campo no exista ningún valor asociado

Todos los campos de la tabla pictures están creados con el atributo Not Null.

Se barajó la posibilidad de añadir la tabla admin, pero con el uso de isAdmin cumpliría la función de tener una tabla extra, y simplificaría mucho el manejo desde el servidor de las peticiones a la base de datos. A medida que se añadieran nuevos servicios a la aplicación habría que ir expandiendo este modelo, resultando muy escalable y fácil de mantener.

Diagrama de clases:

El código estará compuesto por cuatro clases, una contendrá el código para conectar a la base de datos que será usada por las demás clases a través de un atributo que contendrá una instancia de este objeto.

El resto de clases serán una representación en el código de la tabla a la que hacen referencia dentro de la base de datos, sirviendo cada una de ellas para manipular la información de cada una de estas tablas dentro de la aplicación, cada una constará de sus métodos para manipular las tablas pertinentes y tendrán una asociación con la clase MySqlConnection unidireccional. Es decir las clases User, Text y Picture conocen a la clase MySqlConnection pero no al revés, por eso contienen en su interior una instancia de esta última.

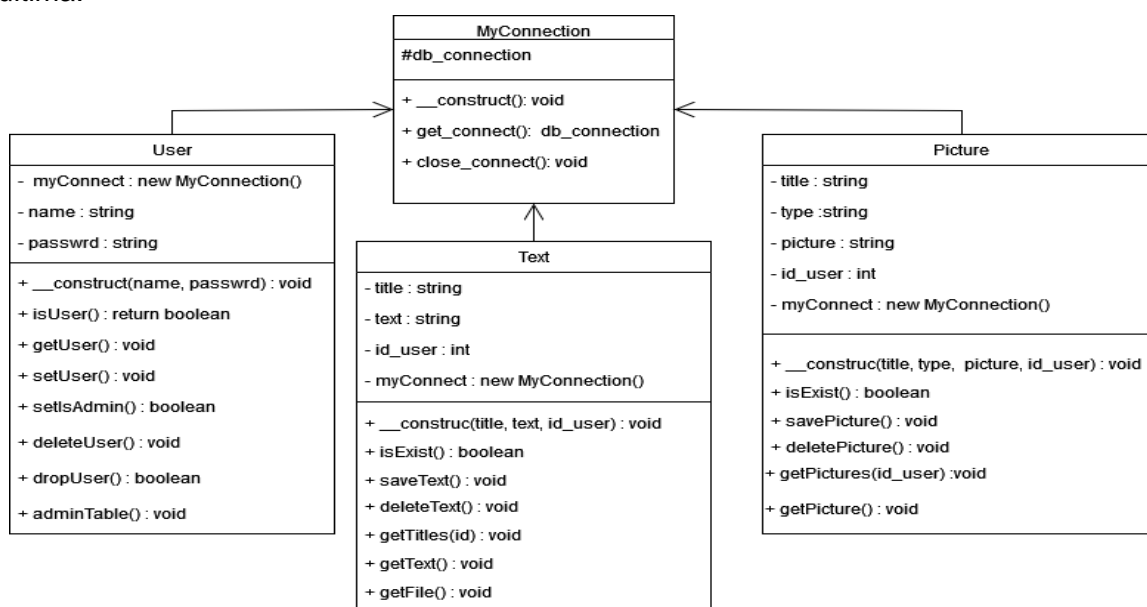


Imagen 5 diagrama de clases (Diagrams.net, 2023)

La aplicación tendrá una estructura de ficheros de MVC en la que archivos de vista, controladores y modelos (clases), estarán separados entre si, y las vistas harán uso de los controladores para interactuar con los modelos que proveerán la información solicitada de

las bases de datos. Haciendo de este un modelo de codificación muy escalable, modular y por tanto, fácil de mantener en el tiempo.

Casos de uso:

A continuación vamos a explicar los casos de uso analizados en esta fase, puede encontrar el diagrama de casos de uso en el Anexo.

Identificador caso de uso	CU 1
Nombre	Entrar en página inicial
Descripción	Actor entra en el menú principal de la aplicación web.
Actores	Admin, User, Invitado
Secuencia Normal	
Actor	Software
Actor introduce la url de la app.	
Excepciones	Devuelve la página inicial de DgToolbox. Software
CU relacionados	CU 2, CU3, CU4, CU5, CU6, CU7
Precondición	
Postcondición	El actor accede a la aplicación a través de su navegador
Identificador caso de uso	CU 2
Nombre	Verificación sesión activa
Descripción	El sistema verifica si hay una sesión activa.
Actores	Admin, User, Invitado
Secuencia Normal	
Actor	Software
Actor entra en la página principal o en alguna de las áreas internas de la app (procesador, galería, área admin).	
Excepciones	Verifica si hay una variable de sesión activa en el navegador. Software
CU relacionados	CU 1, CU 8
Precondición	El actor debe de haber realizado con éxito el CU 1.
Postcondición	Si hay una sesión activa el actor entrara al área de la aplicación con sus credenciales y permisos adecuados a su rol.
Identificador caso de uso	CU 3
Nombre	login
Descripción	El actor iniciara sesión mediante sus credenciales.
Actores	Admin, User
Secuencia Normal	
Actor	Software

Actor pulsa en el menú desplegable de la barra de navegación.

Actor introduce sus credenciales, y pulsa en el botón de envió.

Despliega un menú de registro.

Verifica en el sistema que las credenciales son válidas, y devuelve una variable de sesión con los permisos y datos necesarios del actor.

Excepciones

Credenciales no validas

Software

Devuelve un mensaje de error al actor informando del error.

CU relacionados

Precondición

CU 1

No tener una sesión de usuario activa en el navegador.

Postcondición

Actor recibe sus credenciales y permisos de acceso.

Identificador caso de uso

Nombre

Descripción

CU 4

Registro

Registro de nuevo usuario en la base de datos

Actores

Invitado

Secuencia Normal

Actor

Software

Actor pulsa en el menú desplegable de la barra de navegación.

Despliega un menú de registro.

Actor introduce campos requeridos y pulsa la casilla de new user.

Sistema verifica que el usuario no existe en la base de datos.

Sistema valida los campos y realiza la inserción del registro en la base de datos encriptando la contraseña.

Sistema crea una variable de sesión con las credenciales y permisos del usuario registrador al actor

Excepciones

Usuario ya existe

Software

Sistema devuelve un mensaje específico de error al actor

Error al conectar con la base de datos

Sistema devuelve un mensaje específico de error al actor

CU relacionados

Precondición

CU 1

No tener una sesión activa en el navegador

Postcondición

Invitado obtiene permisos de usuario y queda persistido en base de datos.

Identificador caso de uso

Nombre

Descripción

Actores

CU 5

logout

Cerrar sesión de usuario activa

Admin, User

Secuencia Normal	
Actor	Software
Actor pulsa en el menú desplegable de la barra de navegación.	Despliega un menú de registro.
Actor pulsa botón de logout.	Sistema borra la variable de sesión de usuario del navegador. Sistema devuelve al recurso de la aplicación y recarga con el nuevo estado de sesión.
Excepciones	Software
CU relacionados	CU 1, CU 8, CU14
Precondición	Tener una sesión de usuario activa.
Postcondición	Actor pasara a tener permisos de Invitado.
Identificador caso de uso	CU 6
Nombre	Borrar usuario
Descripción	Borrar usuario propio de la base de datos.
Actores	User
Secuencia Normal	
Actor	Software
Actor pulsa en el menú desplegable de la barra de navegación.	Despliega un menú de registro.
Actor pulsa botón de eliminar usuario.	Sistema borra el registro de usuario de la base de datos que coincide con el número identificador de la sesión activa y todos sus archivos asociados a la cuenta.
Excepciones	Software
Error al conectar con la base de datos	Sistema devuelve un mensaje específico de error al actor.
CU relacionados	CU 1, CU 8, CU14
Precondición	Estar registrado y tener una sesión activa en el navegador
Postcondición	El actor eliminara su cuenta y todo lo persistido junto con ella en la base de datos y será redirigido a la appweb con permisos de Invitado.
Identificador caso de uso	CU 7
Nombre	Mostrar opciones
Descripción	Muestra las herramientas dentro del menú principal a las que tendrá acceso el actor según sus permisos.
Actores	Admin, User, Invitado
Secuencia Normal	
Actor	Software
Actor carga la página principal.	Sistema verifica los permisos que tiene el actor y los botones de acceso a las

Excepciones	herramientas a las cuales tiene permiso. Software
CU relacionados Precondición Postcondición	CU 1 Termina de renderizar la interfaz gráfica con las opciones pertinentes.
Identificador caso de uso Nombre Descripción Actores Secuencia Normal Actor	CU 8 Entrar en la galería de imágenes Actor entra en la galería imágenes de la aplicación web. Admin, User, Invitado
Desde el menú principal el actor pulsa en el botón de galería.	Software
Excepciones	Sistema redirige al actor a la galería de imágenes. Software
CU relacionados Precondición Postcondición	CU2, CU5, CU6, CU10, CU11, CU12, CU13 Actor accede al recurso con las credenciales adecuadas a su rol.
Identificador caso de uso Nombre Descripción Actores Secuencia Normal Actor	CU 9 Visualizar galería Carga una galería de imágenes con las imágenes que el actor tiene persistidas asociadas a su id. Admin, User, Invitado
Actor entra en la galería.	Software
Excepciones	Sistema accede de forma asíncrona a las imágenes persistidas por el actor. Devuelve una renderización en el área reservada al álbum dentro de la galería, con todas las imágenes asociadas. Si no hay imágenes asociadas devolverá un texto "empty". Software
Error al conectar con la base de datos	Sistema devuelve un mensaje específico de error al actor
CU relacionados Precondición Postcondición	CU8 Se podrá interactuar con las fotos mostradas en la galería, y visualizar una vista previa de ellas.

Identificador caso de uso	CU 10
Nombre	Guardar imagen
Descripción	Persistirá una imagen enviada por el usuario en la base de datos
Actores	Admin, User
Secuencia Normal	
Actor	Software
Actor incluirá una imagen en el input habilitado en la galería, y pulsara el botón para enviar.	
Sistema recogerá la imagen de manera temporal y la insertara en su tabla de la base de datos, asociada al id del actor y encriptada. Todo esto de manera asíncrona Al terminar detonara el CU 9	
Excepciones	Software
Error al conectar con la base de datos	Sistema devuelve un mensaje específico de error al actor
CU relacionados	CU 8, CU 9
Precondición	
Postcondición	Aparecerá en el álbum de la galería la imagen añadida a la base de datos

Identificador caso de uso	CU 11
Nombre	Borrar imagen
Descripción	Borrara una imagen de la base de datos, visualizada en la galería.
Actores	Admin, User
Secuencia Normal	
Actor	Software
Actor pulsa sobre menú desplegable de la imagen que desea borrar	
Sistema despliega un menú oculto	
Actor presiona sobre botón de borrar	
Sistema borra de manera la imagen seleccionada de la base de datos, y confirmara el evento con un mensaje. Al terminar detonara el CU 9	
Excepciones	Software
Fallo al borrar la imagen o conectar con la base de datos.	Devolverá un mensaje específico del error.
CU relacionados	CU 9, CU 8
Precondición	Tener almacenada al menos una imagen en la base de datos.
Postcondición	Actualización inmediata del álbum de la galería.

Identificador caso de uso	CU 12
Nombre	Descarga de imagen
Descripción	Actor descargara una de las imágenes

Actores	almacenadas en su equipo
Secuencia Normal	Admin, User
Actor	Software
Actor hará click en la imagen que quiere descargar.	El sistema recuperara la imagen de la base de datos y guardara en el servidor temporalmente. Redirige al actor hacia una página nueva con un botón que hará de enlace de descarga.
El actor pulsara el botón de descarga.	El sistema usara el archivo guardado temporalmente en el servidor y los descargara en el equipo del actor en la carpeta de descargas.
Actor cierra la ventana de descarga.	Sistema borra el archivo temporal del servidor.
Excepciones	Software
CU relacionados	CU 8
Precondición	Tener almacenada al menos una imagen en la base de datos.
Postcondición	Archivo deseado descargado en carpeta descargas del equipo. Archivo temporal eliminado del servidor.
Identificador caso de uso	CU 13
Nombre	Volver al menú inicial
Descripción	Regresa al actor al menú principal
Actores	Admin, User, Invitado
Secuencia Normal	
Actor	Software
Actor pulsara el icono de la aplicación DgToolbox. Situado a la izquierda de la barra de navegación.	Sistema redirigirá al actor al menú inicial de la aplicación.
Excepciones	Software
CU relacionados	CU8, CU14
Precondición	Estar en alguno de los recursos de la aplicación que no sea el menú principal.
Postcondición	Actor redirigido de nuevo al menú principal.
Identificador caso de uso	CU 14

Nombre	Entrar al procesador de texto
Descripción	Actor entra en la galería imágenes de la aplicación web.
Actores	Admin, User, Invitado
Secuencia Normal	
Actor	Software
Desde el menú principal el actor pulsa en el botón de galería.	
	Sistema redirige al actor a la galería de imágenes.
Excepciones	Software
CU relacionados	CU1, CU2, CU5, CU6, CU13, CU16
Precondición	
Postcondición	Actor accede al recurso con las credenciales adecuadas a su rol.
Identificador caso de uso	CU 15
Nombre	Guardar texto
Descripción	Persistirá un texto enviado por el usuario, en la base de datos
Actores	Admin, User
Secuencia Normal	
Actor	Software
El actor dará un título a su documento y pulsará enviar.	
	Sistema recogerá el texto y lo insertará en su tabla de la base de datos, asociada al id del actor y encriptado. Todo esto de manera asíncrona
	Se dispara el CU18
	Devolverá un mensaje de éxito en el procesador de texto.
Excepciones	Software
Error al conectar con la base de datos	Sistema devuelve un mensaje específico de error al actor
CU relacionados	CU14, CU18
Precondición	
Postcondición	Aparecerá en el álbum de la galería la imagen añadida a la base de datos
Identificador caso de uso	CU 16
Nombre	Recuperar textos
Descripción	Recupera un texto de la base de datos y lo plasma en el procesador
Actores	Admin, User
Secuencia Normal	
Actor	Software

Actor pulsa sobre el historial

Se despliega el select que forma el historial de textos

Pulsa sobre una de las entradas

El sistema rellena el campo título con el nombre de la entrada y recupera el texto imprimiéndolo en la ventana del procesador.

Excepciones

Software

CU relacionados

Precondición

CU14

Tener textos persistidos en la base de datos y sesión de usuario activa

Postcondición

Texto recuperado cargado en el procesador.

Identificador caso de uso

Nombre

Descripción

CU 17

Borrar texto

Borrara un texto de la base de datos, visualizada en la galería.

Actores

Admin, User

Secuencia Normal

Actor

Software

Actor pulsa sobre el registro del historial que quiere borrar o escribe el título del archivo en el campo de título y pulsa sobre el botón de borrar

El sistema borrar el archivo de la base de datos persistido con el valor del campo título y asociado al id del actor.

Se dispara CU 18

Excepciones

Software

Fallo al borrar el texto o conectar con la base de datos.

Devolverá un mensaje específico del error.

CU relacionados

Precondición

CU14, CU18

Tener almacenada al menos una imagen en la base de datos.

Postcondición

Actualización inmediata del álbum de la galería.

Identificador caso de uso

Nombre

Descripción

CU 18

Actualiza historial

Historial de textos persistidos se actualiza con los cambios de información.

Actores

Admin, User

Secuencia Normal

Actor

Software

Actor entra con sesión iniciada, guarda o

borra algún texto

Sistema actualiza el contenido del historial en tiempo real y de manera asíncrona.

Excepciones

Software

CU relacionados

Precondición

Postcondición

CU14, CU15, CU17

Historial actualizado con los últimos cambios de los registros del usuario en la tabla texts.

Identificador caso de uso

Nombre

Descripción

CU 19

Descargar texto

Descargar persistido o escrito desde cuenta

Invitado

Actores

Admin, user, Invitado

Secuencia Normal Admin/User

Actor

Software

Actor selecciona texto persistido en el historial, y presiona sobre el botón de descarga emergente.

El sistema recuperara del texto de la base de datos y guardara en el servidor temporalmente.

Redirige al actor hacia una página nueva con un botón que hará de enlace de descarga.

El actor pulsara el botón de descarga.

El sistema usara el archivo guardado temporalmente en el servidor y los descargara en el equipo del actor en la carpeta de descargas.

Actor cierra la ventana de descarga.

Sistema borra el archivo temporal del servidor.

Secuencia Normal Invitado

Actor

Software

Actor escribe un texto en el procesador, pulsa en el botón de descarga emergente.

El sistema recuperara del texto de la base de datos y guardara en el servidor temporalmente.

Redirige al actor hacia una página nueva con un botón que hará de enlace de descarga.

El actor pulsara el botón de descarga.

El sistema usara el archivo guardado temporalmente en el servidor y los descargara en el equipo del actor en la carpeta de descargas.

Actor cierra la ventana de descarga.

Sistema borra el archivo temporal del servidor.

Excepciones	Software
CU relacionados Precondición Postcondición	CU 14 Tener al menos un texto persistido en la base de datos o escribir contenido en el procesador como invitado. Archivo deseado descargado en carpeta descargas del equipo. Archivo temporal eliminado del servidor.
Identificador caso de uso Nombre Descripción Actores Secuencia Normal Actor	CU 21 Escribir texto Admin, User, Invitado
Actor entra en el procesador de texto	Software
Actor redacta el contenido que desea en el procesador.	Sistema carga un procesador funcional en pantalla, con diversas funciones de edición.
Excepciones	Software
CU relacionados Precondición Postcondición	CU14 Se obtiene un texto listo para ser guardado y / o descargado.
Identificador caso de uso Nombre Descripción Actores Secuencia Normal Actor	CU 22 Entrar en área admin Admin
Actor pulsa el botón área admin del menú principal	Software
	Sistema carga un área para gestión de usuarios y permisos.
Excepciones	Software
CU relacionados Precondición Postcondición	CU1,CU2 Tener iniciada sesión como administrador. Se obtiene un texto listo para ser guardado y / o descargado.
Identificador caso de uso Nombre Descripción	CU 23 Eliminar usuarios Elimina usuarios desde el menú

Actores	administrador
Secuencia Normal	Admin
Actor	Software
Actor pulsa el botón de eliminar usuario sobre el registro del usuario que quiere eliminar.	Sistema elimina el usuario elegido y todos sus archivos asociados
Excepciones	Software
Error al modificar un registro	Se informa del error mediante un mensaje de advertencia.
CU relacionados	CU 22
Precondición	Tener iniciada sesión como administrador.
Postcondición	Se modifica exitosamente las tablas de las bases de datos y se actualiza el estado de la tabla del área administrador inmediatamente.
Identificador caso de uso	CU 24
Nombre	Cambiar permisos
Descripción	Cambia los permisos de los usuarios registrados
Actores	Admin
Secuencia Normal	Software
Actor	Software
Actor pulsa el botón de cambios de permiso del usuario sobre el registro del usuario que quiere otorgar o denegar acceso a permisos de administrador.	Sistema modifica el usuario elegido y sus permisos
Excepciones	Software
Error al modificar un registro	Se informa del error mediante un mensaje de advertencia.
CU relacionados	CU 22
Precondición	Tener iniciada sesión como administrador.
Postcondición	Se modifica exitosamente las tablas de las bases de datos y se actualiza el estado de la tabla del área administrador inmediatamente.

7. Diseño del proyecto

Diseño de las Interfaces graficas:

En esta parte del diseño la idea según los requisitos de la fase de análisis era hacer una interfaz moderna y que fuera intuitiva, que hiciera uso de iconos siempre que fuera posible a

lo largo de su desarrollo. Además para esta parte del proyecto elegí utilizar 2 framework que facilitarían su creación, y además, permitirían que su estructura y diseño fuera responsive y se adaptara a cualquier dispositivo y navegador, los cuales fueron Bootstrap y FontAwesome.

Podremos distinguir las interfaces en dos partes que se mantendrán constantes a lo largo del proyecto:

- Barra del navegador

- Cuerpo de la aplicación

El navegador será el mismo durante toda la aplicación y será cargado a través del servidor en función del rol y estado de los permisos a través de una inserción de código php en el código Html. Pero siempre constara de un icono de la app que servirá para poder regresar al menú principal y un menú desplegable a la derecha que servirá para funciones de gestión de usuario y sesión.

El cuerpo de cada recurso de la app variara según su propósito. En el anexo puedes encontrar los wireframes realizados durante esta fase.

Para la tipografía se utilizó fuentes del api googleFonts en específico la fuente Roboto y variantes de esta en cursiva y negrita.

Diseño de las de la lógica de negocio:

Ahora vamos a explicar en detalle cada componente del proyecto desde un punto de vista técnico y de codificación.

Clases:

MyConnection: Esta clase se encargara de proporcionar una conexión a la base de datos a las demás clases de la aplicación, permitiendo realizar conexiones con una simple instancia de MyConnection en el constructor de cada una de las clases, pudiendo ser utilizado por cada uno de sus métodos al ser llamados desde una instancia de la clase en los controladores, permitiendo una reutilización de código óptima.

La Clase MyConnection cuenta con 2 métodos, y un constructor. El constructor inicia una conexión con la base de datos y la asigna a su única variable, `db_connection`, que será la que se entregue a sus instancias mediante el método `get_connect()`. Y el método `close_connect()` que cerrara la conexión almacenada en la variable `db_connection`.

User: Esta clase representa el enlace entre la tabla users y la lógica de la aplicación, en ella tendremos todo lo necesario para interactuar con la tabla users, ya sea para extraer, introducir o modificar información. Se encargará también del cifrado de contraseñas.

Consta de los atributos name, passwrđ y myConnect, este último será la instancia de Myconnection dentro de la clase User. Por otro lado, name y user se rellenarán según convenga con la información proveniente del controlador.

Después tenemos el constructor que inicializara los atributos con la información de la instancia del controlador y con la instancia de MyConnection anteriormente explicada.

Tenemos los siguientes métodos:

- **isUser():** Este método verificara si el usuario que corresponde a la instancia existe en la base de datos devolviendo un booleano por respuesta.
- **getUser():** Este método hace uso de isUser() para determinar si los datos proporcionados coinciden con algún registro en la base de datos, si es así localiza el registro y puebla una variable de sesión con los datos del usuario (id, name ,isAdmin).Esto lo consigue llenando la variable con un array asociativo que será de gran utilidad a la hora de comprobar en una sola variable el id, nombre y permisos del usuario logeado. Sin embargo si los datos no coinciden con ningún usuario o se produce algún error de conexión, devolverá el mensaje de error específico al a la vista del usuario a través de la url.
- **setUser():** Este método hace uso de isUser() para determinar si los datos proporcionados coinciden con algún registro en la base de datos, si es así, o si ocurre un error en la conexión devolverá un mensaje de error específico al problema hacia el controlador. Si no se haya ningún registro coincidente se insertara un nuevo registro en la base de datos con los datos proporcionados y encriptando la contraseña con cifrado AES (solingest, 2023), y posteriormente se comprobara mediante una query¹⁷ que el registro quedo persistido, si se corrobora con éxito se hará uso de la función getUser() para realizar el login automáticamente. De lo contrario se informará del error mediante un mensaje a la vista del usuario a través de la url.
- **setIsAdmin():** Este método es usado en el panel de control del administrador y servirá para cambiar los permisos a un usuario concediéndole el rol de administrador o revocándose y devolverá true al controlador. En caso de error con la base de datos devolverá false.

¹⁷ Peticiones que se realizan a la base de datos.

- **deleteUser():** Este método servirá para borrar un usuario desde el menú desplegable del navegador y servirá únicamente para borrar la cuenta con la que se ha iniciado la sesión. Una vez efectuada la conexión se procederá a borrar todos los textos y todas las imágenes asociadas al id del usuario, posteriormente se borrará el usuario evitando así incoherencias en la base de datos. En caso de error en la conexión con la base de datos devolverá un mensaje de error a la vista del usuario a través de la url.
- **dropUser():** Este método permitirá al administrador borrar los usuarios desde el panel de control donde visionará los registros de la base de datos, borrando primero los archivos asociados a este y finalmente el usuario de la tabla users, al acabar devolverá true de lo contrario y en caso de error de conexión devolverá false.
- **adminTable():** Este método cargará todos los registros de la tabla users en un resultset y volcará los datos en una tabla que dispondrá de los botones necesarios para procesar las acciones por javascript en la vista y los devolverá como un string al controlador. En caso de error devolverá un mensaje de al controlador informando de la causa.

Text: Esta clase representa el enlace entre la tabla texts y la lógica de la aplicación, en ella tendremos todo lo necesario para interactuar con la tabla texts, ya sea para recuperar, introducir o modificar información. Se encargará también del cifrado de los archivos de texto.

Estará conformada con los atributos title, text, id_user que se rellenarán según convenga con la información proveniente del controlador, y finalmente el atributo myConnect que se usará para la instancia de MySqlConnection.

Después tenemos el constructor que inicializará los atributos con la información de la instancia del controlador y con la instancia de MySqlConnection anteriormente explicada. Además, todos estos métodos serán operados de manera asíncrona mediante AJAX.

Tenemos los siguientes métodos:

- **isExist():** Este método verificará si existe en la base de datos algún registro que coincida con los campos id_user y title devolviendo un booleano por respuesta.
- **saveText():** Este método verificará haciendo uso de isExist() si existe el texto a guardar previamente en la base de datos si devuelve true, entonces realizará un update¹⁸ del campo text de la tabla sobrescribiendo el contenido del campo. Y si isExist() devuelve false realizará una inserción normal en la tabla texts persistiendo

¹⁸ Tipo de query que trata de actualizar datos en la base de datos.

el nuevo texto y devolviendo al controlador una respuesta de “success”. Si se produce algún error en el proceso devolverá un error al controlador devolviendo “fail”, estos mensajes los aprovechara el fichero js para la response de Ajax y todas las inserciones del campo text irán encriptadas con cifrado AES.

- **deleteText():** Accede a la tabla text de la base de datos y borra el texto usando los atributos id_user y title para la query. Finalmente devuelve “success” al controlador o “fail si ocurre algún error en la conexión o procedimiento”.
- **getTitles():** Este método es estático y recibe el parámetro id del usuario. Se encarga de cargar la información del desplegable de historial de textos de la aplicación. Recuperará todos los títulos de los textos asociados al usuario que tiene iniciada la sesión. Y renderizará las etiquetas option del menú select. En caso de error de conexión devolverá el mensaje de error al controlador.
- **getText():** Este método recupera el texto seleccionado en el historial de textos de la aplicación , utilizando el id_user y title, ubica el archivo a recuperar, escribe el contenido del texto y lo devuelve al controlador. En caso de error de conexión o en el procedimiento devolverá el string “error”. Estas cadenas de caracteres se trataran desde js con AJAX de manera asíncrona.
- **getFile():** Este método recupera un texto usando los atributos id_user y title para posteriormente devolver el blob correspondiente al texto al controlador donde será tratado. Se utilizará al descargar un texto de la base de datos, el cual se guardara de manera temporal en el servidor. En caso de error de conexión o en el procedimiento devolverá el string “error”.

Picture: Esta clase representa el enlace entre la tabla pictures y la lógica de la aplicación, en ella tendremos todo lo necesario para interactuar con la tabla pictures, ya sea para recuperar, introducir o modificar información. Se encargará también del cifrado de los archivos de imagen. Además, todos estos métodos serán operados de manera asíncrona mediante Ajax.

Estará conformada con los atributos title, type, picture, id_user que se rellenaran según convenga con la información proveniente del controlador, y finalmente el atributo myConnect que se usara para la instancia de MyConnection.

Después tenemos el constructor que inicializara los atributos con la información de la instancia del controlador y con la instancia de MySqlConnection anteriormente explicada.

Tenemos los siguientes métodos:

- **isExist():** Este método verificara si existe en la base de datos algún registro que coincida con los campos id_user y title devolviendo un booleano por respuesta.
- **saveText():** Este método verificara haciendo uso de isExist() si existe la imagen a guardar previamente en la base de datos si devuelve true, entonces realizara un update del campo picture de la tabla sobrescribiendo el contenido del campo. Y si isExist() devuelve false realizará una inserción normal en la tabla picture persistiendo la nueva imagen y devolviendo al controlador una respuesta de “success”. Si se produce algún error en el proceso devolverá un error al controlador devolviendo “fail”, todas las inserciones del campo text irán encriptadas con cifrado AES.
- **deletePicture():** Este método accede a la tabla picture de la base de datos y borra la imagen usando los atributos id_user y title para la query. Finalmente devuelve “success” al controlador o “fail si ocurre algún error en la conexión o procedimiento”.
- **getPictures():** Este es un método estático que recibe el parámetro del id del usuario que tiene una sesión activa y recupera todos los archivos de imagen persistidos en la tabla pictures y lo plasma en el álbum de la galería de la aplicación con sus menús desplegables listos para ser utilizados por los ficheros de javascript para darles funcionalidad. Si no encuentra registros devolverá el string “empty” que se plasmará en el álbum. Si por el contrario hay un error de conexión se enviara un mensaje de error al usuario.
- **getPicture():** Este método recupera una imagen usando los atributos id_user y title para posteriormente devolver el blob correspondiente al texto al controlador donde será tratado. Se utilizará al descargar una imagen de la base de datos, la cual se guardará de manera temporal en el servidor. En caso de error de conexión o en el procedimiento devolverá el string con el mensaje de error adecuado.

Ficheros JavaScript y controladores:

index.js: Este archivo integrara toda la lógica front-end con index.php que será el fichero con la vista correspondiente al menú principal. Este fichero llama a unos gestores de eventos cuando el contenido de la vista se haya cargado, todos ellos asociados al id de cada uno de los botones del menú principal. Que cada uno llamara a su función correspondientes que serían: toProcesador(), toGalery(), toIndex() y toAdminArea(). Todos estos métodos realizan lo mismo en su interior contienen la misma sentencia cambiando el contenido del string que indica el fichero al cuál dirigirse. Usando la sentencia `window.location = "url al fichero de la vista";`

procesador.js: Este archivo integrará toda la lógica front-end con procesador.php que será el archivo con la vista correspondiente al procesador de texto de la aplicación. En primer lugar después de cargar todo el contenido de la vista, antes de cargar los gestores de eventos y otras sentencias. Este archivo dada su complejidad vamos explicar más detenidamente sus partes:

Funciones Ajax

- **getHistory():** Este método realiza una petición post mediante Ajax al controlador history.php, que utiliza el método estático getTitles() de la clase Text y devuelve su resultado como respuesta a la petición Ajax y esta a su vez al método responseGetHistory().
- **saveAjax():** Este método valida que el título de la imagen que recibe sea correcto en caso de no pasar los criterios se escribe un mensaje alert en la vista indicando el error y se devuelve false cortando el flujo de ejecución, si se cumple con la validación lo envía junto con el contenido del texto mediante una petición post con Ajax al controlador saveText_controller.php, Este controlador recoge los datos junto al id de usuario asociado a la variable de sesión, y genera una instancia de la clase Text y utiliza el método saveText(), y finalmente devuelve el resultado como respuesta a la petición Ajax y esta a su vez a responseSaveText().
- **deleteAjax():** Este método valida que el título de la imagen que recibe sea correcto en caso de no pasar los criterios se escribe un mensaje alert en la vista indicando el error y se devuelve false cortando el flujo de ejecución, si se cumple con la validación, envía la variable title al controlador deleteText.php, este controlador

recupera el id del usuario que tiene iniciada la sesión de la variable de entorno session del navegador y usa el método deleteText() de la clase Text. Al terminar se dirige a la respuesta de la petición Ajax que es el método responseDeleteText().

- **historyAjax():** Este método se dispara cuando ocurre un cambio en la selección del historial de textos, al dispararse recoge el título del campo escogido y lo coloca automáticamente en el input usado para ubicar el título del archivo. Esto se hace para verificar el título del texto a recuperar por parte del usuario y para facilitar el uso de las funciones de guardar y borrar, así es más difícil que cometa error al escribir el campo al borrar o querer modificar el archivo. Hecho este paso conecta mediante una petición post con Ajax al controlador getText.php, justo con la información correspondiente al nombre del archivo de texto. Este controlador recupera el id del usuario que tiene iniciada la sesión de la variable de entorno session del navegador y usa el método getText() de la clase Text, Finalmente se dirige a la respuesta de la petición Ajax que es el método write().

Funciones:

- **setLink() y setGuestLink():** estos dos métodos se encargan de generar el enlace a la página de descarga cuando se selecciona un texto a recuperar o cuando el invitado presione una tecla teniendo el foco en el procesador así podrá escribir y descargar lo que tenga hecho en ese momento.
- Función anónima línea 21: sirve para redirigir al usuario hacia el menú principal cuando se pulsa sobre el logo de la aplicación, situado en la barra de navegación.
- Funciones anónimas líneas 26-40: se encargan de animar los iconos de Bootstrap.

Respuestas Ajax:

- **responseSaveText() y responseDeleteText():** ambos funcionan de la misma manera reciben una respuesta del controlador y generan un mensaje alert de Bootstrap en la vista del cliente informando de cómo se resolvió la petición.

Write(): recibe la respuesta del controlador getText.php y escribe los datos recibidos en el área del procesador.

- **responseGetHistory():** recibe la respuesta del controlador history.php, y genera todas las opciones del menú historial de textos, acorde con los registros de textos persistidos por el usuario.

Procesador de texto:

Al inicio del documento procesador.js como se puede observar se le atribuye al textArea con id “userText” una función que es Editor(). Esta lo que hace es desplegar en esa área de texto todo el plugin que usara la aplicación para generar el procesador de texto. El plugin cuenta con una licencia de BY, el plugin se llama LineControl 1.1.0 y se le atribuye a Suyati Technologies (Suyati Technologies, 2014), también fue mencionado en el código tal como venía en su web, de la siguiente manera:

LineControl 1.1.0

* Copyright (C) 2014, Suyati Technologies

* This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

El software fue modificado para este caso en la línea 741 debido a un enlace mal actualizado a los iconos de FontAwesome.

Además del método Editor(), mencionado anteriormente se utiliza para recuperar el texto en saveAjax() mediante Editor(“getText”) cuyo código est en la línea 1588 del archivo editor.js. y en el método de respuesta Ajax, write(), con Editor(“setText”,{texto}), cuyo código está en la línea 1596 del archivo editor.js.

Todo el código se puede encontrar en los archivos de **editor.js** y **editor.css**.

galery.js: Este archivo integrará toda la lógica front-end con galery.php que será el archivo con la vista correspondiente a la galería de imágenes de la aplicación. En primer lugar después de cargar todo el contenido de la vista, antes de cargar los gestores de eventos y otras sentencias. Este archivo dada su complejidad vamos explicar más detenidamente sus partes:

Funciones Ajax:

- **getPictures():** Este método hace una petición post mediante Ajax al controlador paintGalery.php, este controlador utiliza el método getPictures() de la clase Picture pasándole el id del usuario con sesión activa, después la respuesta es utilizada en la respuesta Ajax responsePaintGalery().
- **deletePic():** Este método obtiene el título de la imagen a la que corresponde el botón de borrado sobre el que se hace click, que es el evento que desencadena su llamada, este se guarda en un atributo id del botón. Posteriormente hace una petición post mediante Ajax al controlador deletePic_controller.php pasándole el valor title. Este controlador obtiene como en los casos anteriores el id del usuario con sesión activa y usa el método deletePicture() de la clase Picture, finalmente se devuelve el flujo a la respuesta Ajax deleteResponse().
- **uploadAjax():** Este método recoge la imagen seleccionada en el input file de la vista y la envía mediante una petición post con Ajax al controlador uploadPic_controller.php este se encargará de obtener la imagen guardada temporalmente en la variable FILES del navegador con todos sus datos necesarios para ser usados luego en el método savePicture() de la clase Picture. Previamente pasará por una validación para corroborar que el archivo enviado es uno de los permitidos para ser guardado en nuestra aplicación de lo contrario devolverá un valor de error. Finalmente el flujo volverá a la respuesta Ajax uploadResponse().

Respuestas Ajax:

- **responsePaintGalery():** Dibuja el álbum con la información recuperada de la petición a paintGalery.php.
- **uploadResponse() y deleteResponse():** dependiendo del valor devuelto por el controlador informarán mediante un alert de Bootstrap de cómo se ha resuelto la petición en la vista del cliente.

Funciones: encontramos una función `toIndex()` como en `index.js` que devuelve al menú principal pulsando sobre el icono de la aplicación. Y los métodos `iconSpin()` y `iconStop()` que se encargaran de gestionar las animaciones de los iconos, iniciándolos y parando la animación respectivamente mediante el uso de atributos de clase.

downloadPage.js: Este archivo integrará toda la lógica front-end con `downloadPage.php` que será el archivo con la vista correspondiente al enlace de descarga de la aplicación. Este archivo javascript solo tiene una función anónima que se dispara a cerrar la pestaña y antes de que esto se lleve a cabo con un evento `beforeunload` el cual eliminara de la carpeta `assets` del servidor el archivo que hemos guardado temporalmente para que el cliente pueda descargarlo en su equipo.

adminArea.js: Este archivo integrará toda la lógica front-end con `adminArea.php` que será el archivo con la vista correspondiente a el área de administrador de la aplicación. En primer lugar después de cargar todo el contenido de la vista, antes de cargar los gestores de eventos y otras sentencias. Este archivo dada su complejidad vamos a explicar más detenidamente sus partes:

Funciones Ajax:

- **adminTable():** Este método se utilizara para crear la tabla de datos del administrador se disparara cada vez que se cargue el archivo `adminArea.php`. Realizará una petición post Ajax al controlador `adminTable.php` este a su vez usara el metodo estático `adminTable` de la clase `User`. Finalmente el flujo volverá a la respuesta Ajax `drawTable()`.
- **dropUser():** Este método realizará una petición post al controlador `dropUser.php` enviándole el id del usuario a borrar, en el controlador borrara el usuario mediante la función `dropUser` de la clase `User` y como devuelve un booleano dependiendo de lo que devuelva si sale con éxito volverá a obtener los datos la tabla de administrador sin ese registro o acabara con la ejecución en caso de error y no modificaría la tabla. Finalmente si tiene éxito el flujo volverá a la respuesta Ajax `drawTable()`.
- **changeAdmin():** Este método funciona parecido al anterior recupera el id del usuario seleccionado y se lo pasaría a la petición post Ajax hacia el controlador

changeAdmin.php. En el controlador cambiara los permisos del usuario mediante la función `setIsAdmin()` de la clase `User` y como devuelve un booleano dependiendo de lo que devuelva si sale con éxito volverá a obtener los datos la tabla de administrador sin ese registro o acabara con la ejecución en caso de error y no modificaría la tabla. Finalmente si tiene éxito el flujo volverá a la respuesta Ajax `drawTable()`.

La única respuesta Ajax que existe en este archivo es común a todos los métodos anteriores siendo `drawTable()` este método con los datos recibidos dibuja la tabla de administrador en el lugar indicado.

Estructura de ficheros:

El proyecto se estructuro en módulos que corresponden con el modelo vista controlador (MVC) y fue la siguiente:

- BBDD: contiene una exportación del modelo de la base de datos.
- src: contiene todo el proyecto con las carpetas que vienen a continuación.
- controller: Archivos php que comunican los archivos javascript de la vista con los modelos en php.
- model: Archivos php que contienen la lógica y las herramientas para interactuar con la base de datos y sus tablas.
- view: Archivos php con las vistas html y archivos css y javascript necesarios estos últimos estarán anidados en sus respectivas carpetas "css" y "js".
- assets: carpeta donde se alojarán recursos temporales como los archivos a descargar en `downloadPage.php`.

8. Despliegue y pruebas

Para el despliegue de la aplicación y poder verificar su desarrollo y ejecutar pruebas de funcionamiento y validación utilicé como expliqué en el apartado de tecnologías del proyecto WAMP. Para que el servidor local pueda desplegar el proyecto requiere de un paso muy sencillo y es pasar la carpeta del proyecto a la carpeta "www", ubicada dentro de la carpeta wamp que se encontrara donde se destinase en la instalación. Una vez depositado el proyecto dentro bastara con escribir en el navegador `http://localhost/{nombre del proyecto}/{recurso}`.

También habrá que importar el script de sql en MyphpAdmin, herramienta de wamp, una vez dentro usaremos el usuario root que por defecto no tiene contraseña, y crearemos una nueva base de datos llamada db_proyecto_daw (como el título del script sql) con la codificación utf8_general_ci, y seguido pulsaremos en la opción **import**, y rellenaremos el input que nos pide el script .sql y pulsaremos en importar abajo del formulario.

Pruebas:

Nº1 Inicio por defecto y acceso como invitado (RF 1, 2 ,6): Se introduce el path hacia el menú principal de la aplicación y se comprueba que inicia por defecto como invitado (Guest). Y se prueban los botones de enlace a las herramientas habilitadas a un invitado, en ellas se observa que las acciones están limitadas a los privilegios que debe tener un invitado en la aplicación, escribiremos un texto con el invitado y corroboraremos que permite descargarlo y que finalmente ese archivo es depositado en la carpeta de descargas de nuestro equipo.

Nº2 Prueba de sistema de login y borrado de usuario (RF 3, 4, 5 y 15):

Parte 1: Se introduce datos en el formulario de login ubicado en la barra de navegación con el campo new user marcado, se comprueba que al enviar genera la sesión y queda guardado en la base de datos mediante phpMyAdmin de WAMP.

Después comprobaremos que la sesión se mantiene activa al cerrar las pestañas y volverlas a abrir y seguido se procede a hacer logout y volver a loguearse si esto tiene éxito finalmente se borra el usuario con el botón de delete user y se comprueba que la sesión haya quedado borrada recargando la página y que el registro se borró en la base de datos.

Parte 2: Esta segunda parte de la prueba habrá que repetirla una vez hayamos verificado las pruebas 3 y 4. Repetiremos los pasos de la parte anterior comprobando que el borrado del usuario se realiza correctamente y también de los archivos asociados a él en las tablas texts y pictures.

Nº3 Prueba de sistema de manipulación de textos por usuario (RF 7, 16): Se introduce un usuario y se prueba a persistir un par de textos ,recuperarlos del historial comprobando que se actualiza correctamente y borrar un par de ellos corroborando el buen funcionamiento del historial y que se recuperan los textos correctamente, siguiendo con la descarga de al menos un fichero, comprobando que se descarga el contenido correcto y que el archivo temporal es borrado de la carpeta assets al cerrar la ventana de descarga.

Nº4 Prueba de sistema de manipulación de imágenes por usuario y funcionamiento de álbum (RF 8, 9 y 16): Ingresaremos con un usuario en la galería de fotos y persistiremos

unas imágenes comprobando que nos devuelve un mensaje confirmándolo y que inmediatamente se cargan en el álbum pudiendo visualizarlas de manera ordenada sin solapamientos. Comprobaremos que si introducimos un formato inválido nos devuelve un mensaje de error informando y capturando la excepción.

Después comprobaremos que podemos borrar alguna de las fotos del álbum y actualizándose este automáticamente. Finalizando con la descarga de algunos de los archivos corroborando que se descargan correctamente y queda borrado el archivo temporal de la carpeta assets al cerrar la ventana de descarga.

Nº5 Pruebas de datos en base de datos (RF 10 y 11): En esta parte accederemos a PhpMyAdmin y comprobaremos visualizando los datos de las tablas que tanto las contraseñas como las imágenes y textos son guardados de forma correcta y encriptados.

Nº6 Pruebas sobre el área de administrador (RF 12, 13, 14, 16): En esta prueba ingresaremos en la aplicación haciendo login con el usuario administrador definido en la base de datos desde un inicio. Comprobaremos que aparece el botón de área admin oculto para el resto de usuarios y que nos lleva al recurso adecuado cargando una tabla de todos los usuarios persistidos en base de datos. Seguido comprobaremos que tanto en la tabla como en el desplegable de la barra de navegación no nos permite borrarse a sí mismo el usuario por razones de seguridad. Y posteriormente que permite cambiar los permisos y borrar usuarios de la base de datos, actualizando la tabla de manera inmediata, verificando a su vez en phpMyAdmin que se realiza de manera satisfactoria y coincide con el estado real de la base de datos.

Por ultimo podemos verificar que si el admin entra al procesador o galería tampoco le permite borrarse a sí mismo desde el desplegable de la barra de navegación. Y que un admin solo puede ser borrado por otro administrador, previamente nombrado por este u otro admin.

9. Conclusiones

Al finalizar este proyecto se puede concluir que los objetivos marcados en un inicio en la propuesta han sido logrados por completo.

Se ha creado una interfaz amena e intuitiva para el usuario, con el uso de iconos, animaciones y botones emergentes que guían la vista y atención del usuario acompañándole por el recorrido de la aplicación, con un diseño que se visualiza bien independientemente del navegador utilizado.

Funcionamiento asíncrono en las herramientas de la aplicación, lo cual es muy beneficioso en el procesador ya que al guardar o borrar textos de base de datos no se pierde el área de trabajo actual como al interactuar con el historial que permite tener lo actualizado sin refrescar la página, como en la galería que podemos borrar y añadir imágenes viendo como nuestro álbum evolucionar de manera inmediata viendo en tiempo real cual es el estado de nuestros archivos en todo momento dentro de la base de datos. Y lo mismo ocurre con el área de administrador. Y poder descargar los archivos en el equipo que es en mi opinión una funcionalidad básica ya que de nada sirve guardar datos y archivos si no puedes recuperarlos de una manera sencilla.

La seguridad es uno de los puntos fuertes en esta aplicación ya que encripta toda la información sensible que el usuario pudiera guardar en nuestros servidores con unos de los cifrados más usados para este tipo de servicios a día de hoy.

En conclusión me encuentro orgulloso del funcionamiento de la aplicación y más aún de cómo he acabado desarrollándome a la hora de gestionar incógnitas y problemas durante su proceso, así como de los conocimientos que he ganado en el proceso: encriptación de datos en base de datos, peticiones asíncronas usando Ajax, manejo de variables globales del lado servidor, transferencia de archivos entre cliente y servidor y viceversa, encabezados de archivos, validación de datos (data binding), construcción de interfaces de usuario con frameworks, capacidad de comprensión de lectura de código ajeno(funcionamiento del plugin y comprensión de usar sus herramientas), comprender los procesos de desarrollo de un software, entre otros puntos que han mejorado mis capacidades como desarrollador de una manera cualitativa. Y que seguro me ayudará a afrontar mi futuro profesional.

10. Vías futuras

Este apartado lo voy a dividir en dos partes la primera hablando de mejoras pendientes para las funcionalidades actuales y escalabilidad de estas y por otro lado hablare sobre funciones no desarrolladas aún pero que deberían ser el siguiente paso hacia la evolución y mejora de la idea planteada en el proyecto.

En el procesador habría que conseguir que podamos descargar los textos en formato .doc ya que si lo hacemos actualmente nos da problemas con la codificación que no me dio tiempo de ajustar y solo podemos descargarlo en formato odt. También habría que desarrollar un paginador tanto para la galería como para la tabla del área de administrador

para evitar y gestionar bien un overflow¹⁹ de registros en la tabla o de imágenes en el álbum.

A día de hoy la parte de login se hace de manera síncrona mediante formularios a php, se intentó desarrollar en Ajax pero se me hecho el tiempo encima y ocurría un error poco común y al enviar la petición de login se hacía correctamente pero al pulsar una opción entre el procesador o galería, no pude asegurar la razón pero se disparaba el método de borrar usuario, finalmente se decidió dejar como estaba ya que es un cambio menor que tampoco traía ninguna mejora especial pero me gustaría llegar al fondo del asunto.

Por otro lado como mejoras a futuro me gustaría implementar un marketplace para que la comunidad pueda añadir funcionalidades a la aplicación libremente pasando por una inspección previa de nuestra empresa, punto del que hablaba en el estado del arte analizando las oportunidades frente a las amenazas que eran la competencia actual y más específicamente google drive tenía en funcionamiento pero no tan explotada como podría estar. Asegurando de realizarse esta parte un crecimiento de las funcionalidades exponencial. Además, habría que añadir cifrado en la transferencia mediante un protocolo de seguridad “https” para adecuarse a los estándares fijados por la competencia ya que el cifrado en reposo ya lo hemos cumplido. También se debería añadir una galería para videos o poder utilizar la actual para poder persistir archivos de video, creo que este sería un paso lógico después de permitir guardar documentos escritos e imágenes y que deberíamos desarrollar el equipo o desarrolladores de DgToolbox y no la comunidad, garantizando la seguridad como hasta ahora.

Finalmente, si se quiere llevar este proyecto a un plano profesional y no solo académico, habría que realizar todo un proceso de gestión para dar de alta la empresa y empezar a contratar empleados para formar un equipo que ayuden a levantar este proyecto, conseguir financiación, elaborar plan de empresa, conseguir un dominio web donde colgar nuestra app (Hosting), etc. Todo lo necesario para que este proyecto pudiera ver la luz.

11. Bibliografía/Webgrafía

Bootstrap. (s.f.). *documentacion Bootstrap*. Recuperado el 25 de 09 de 23, de <https://getbootstrap.esdocu.com/>

Canva. (2023). *Plantilla para diagrama de Gantt*. Obtenido de <https://www.canva.com/>

¹⁹ Desbordamiento de elementos en un documento html al ser renderizado por un navegador.

- Diagrams.net. (2023). *Appweb para la creacion de diagramas*. Obtenido de <https://app.diagrams.net>
- Dropbox. (2023). *About DropBox*. Recuperado el 27 de 09 de 2023, de https://aem.dropbox.com/cms/content/dam/dropbox/www/en-us/about/company-info/DBX_FactSheet.pdf
- Dropbox. (08 de 08 de 2023). *Cómo protege Dropbox tus archivos*. Recuperado el 27 de 09 de 2023, de <https://help.dropbox.com/es-es/security/how-security-works>
- FontAwesome. (25 de 09 de 2023). *doc FontAwesome*. Obtenido de <https://fontawesome.com/docs>
- Google. (2023). *google fonts-roboto*. Obtenido de <https://fonts.google.com/specimen/Roboto?query=roboto>
- Google Drive. (2023). *Empezar a utilizar archivos cifrados en Drive, Documentos, Hojas de cálculo y Presentaciones*. Recuperado el 27 de 09 de 23
- Ilerna online S.L. (2021). Descripción del termino Back-End y Front-End. En I. o. S.L, *Desarrollo web entorno servidor* (pág. 94). Ilerna online S.L.
- Ilerna Online S.L. (2021). Mecanismos de comunicación asíncrona. En I. O. S.L, *Desarrollo web entrono cliente* (pág. 123). Ilerna Online S.L.
- Ilerna Online S.L. (2021). Modelo en cascada. En I. O. S.L, *Entorno de desarrollo* (págs. 25-26). Ilerna Online S.L.
- Ilerna Online.S.L. (2021). Fases de desarrollo. En I. Online.S.L, *Entorno de desarrollo* (págs. 32-51). Ilerna Online.S.L.
- IONOS. (11 de 03 de 2019). *El modelo en cascada*. Obtenido de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- Microsoft. (2023). *¿Qué es el almacenamiento en la nube?* Recuperado el 27 de 09 de 2023, de <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-cloud-storage>
- Microsoft. (2023). *Cómo protege OneDrive los datos en la nube*. Recuperado el 27 de 09 de 2023, de <https://support.microsoft.com/es-es/office/c%C3%B3mo-protege-onedrive-los-datos-en-la-nube-23c6ea94-3608-48d7-8bf0-80e142edd1e1>
- Mocups. (2023). *app web para el diseño de mocups*. Obtenido de <https://my.moqups.com>

News Center Microsoft Latinoamérica. (10 de 08 de 2022). *OneDrive cumple 15 años de entregar experiencias nuevas e intuitivas*. Recuperado el 27 de 09 de 2023, de <https://news.microsoft.com/es-xl/onedrive-cumple-15-anos-de-entregar-experiencias-nuevas-e-intuitivas/>

OpenJS Foundation. (2023). *JQuery Doc*. Obtenido de <https://api.jquery.com/>

OpenJS Foundation. (2023). *What is JQuery?* Obtenido de <https://jquery.com/>

PHP. (2 de agosto de 2023). *Documentacion oficial*. Obtenido de <https://www.php.net/manual/es/preface.php>

solingest. (2023). *Encriptacion AES*. Obtenido de <https://www.solingest.com/almacenar-contrasenas-en-mysql>

Suyati Technologies. (2014). *LineControl 1.1.0 Plugging*. Obtenido de <http://suyati.github.io/line-control>

UNIR. (23 de 12 de 2020). *Qué es wireframing*. Recuperado el 08 de 10 de 2023, de <https://www.unir.net/ingenieria/revista/que-es-wireframing/>

w3schools. (2023). *HTML Doc*. Obtenido de <https://www.w3schools.com/html/default.asp>

wikipedia. (10 de 03 de 2023). *Advanced Encryption Standard*. Obtenido de https://es.wikipedia.org/wiki/Advanced_Encryption_Standard

wikipedia. (23 de junio de 2023). *CSS*. Obtenido de <https://es.wikipedia.org/wiki/CSS>

wikipedia. (16 de enero de 2023). *Document object model*. Obtenido de wikipedia: https://es.wikipedia.org/wiki/Document_Object_Model

wikipedia. (13 de junio de 2023). *ECMAScript*. Obtenido de <https://es.wikipedia.org/wiki/ECMAScript>

wikipedia. (17 de 05 de 2023). *Framework*. Obtenido de <https://es.wikipedia.org/wiki/Framework>

Wikipedia. (23 de 09 de 2023). *Goggle Drive*. Recuperado el 27 de 09 de 2023, de https://es.wikipedia.org/wiki/Google_Drive

wikipedia. (junio de 2023). *JavaScript*. Obtenido de wikipedia: <https://es.wikipedia.org/wiki/JavaScript>

www.DeepL.com. (s.f.). *traductor español-inglés*. Recuperado el 25 de 09 de 2023, de www.DeepL.com/Translator

12. Anexos

12.1. Glosario

AES: Advanced Encryption Standard, es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos, creado en Bélgica (wikipedia, 2023)., 31; Advanced Encryption Standard, es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos, creado en Bélgica., 7, 15, 17, 31, 33, 34

array: es una matriz donde se guarda informacion en el contexto de la programación de software, 31

backend: es lo que se entiende como programación del lado del servidor. Los programadores se encargarán de desarrollar toda la lógica de la aplicación que procesará todos los datos y los almacenará en la base de datos. (Ilerna online S.L, 2021), 3

diagramas E-R: Diagramas utilizados en bases de datos que relacionan las diferentes tablas entre si , mediante el uso de campos clave que son utilizados como referencias., 10

DOM: Es una interfaz de plataforma para programación, o una API diseñada para documentos HTML, que otorga una selección de objetos estándar para simbolizar publicaciones HTML, XHTML, XML y SVG, un patrón de referencia para reunir esos objetos, y una forma de acceso estándar para tener acceso a ellos y manipularlos. (wikipedia, 2023), 11

ECMAScript: Es una especificación para lenguajes de programación desarrollada por la organización Ecma International. Está basado en el popular lenguaje JavaScript propuesto como estándar por Netscape Communications Corporation. Actualmente está aceptado como el estándar ISO/IEC 22275, 10

framework: Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. Para llevar a cabo estas soluciones se sirven de bibliotecas de código específicas para ese uso. (wikipedia, 2023), 12

front-end: Conlleva el desarrollo de todos los aspectos relacionados con el diseño de la aplicación web, la creación de todas las interfaces de los sitios web del dominio. (Ilerna online S.L, 2021), 35

wireframes: es el boceto de cómo serán cada una de las pantallas de una web o app. Pero este diseño no se especifica tipografía ni colores, lo que se busca no es el aspecto visual, sino cuál es el orden, y que estructura tendrá cada una de las ventanas , organización, orientación de los botones. (UNIR, 2020), 30

12.2. Diagrama de Casos de uso

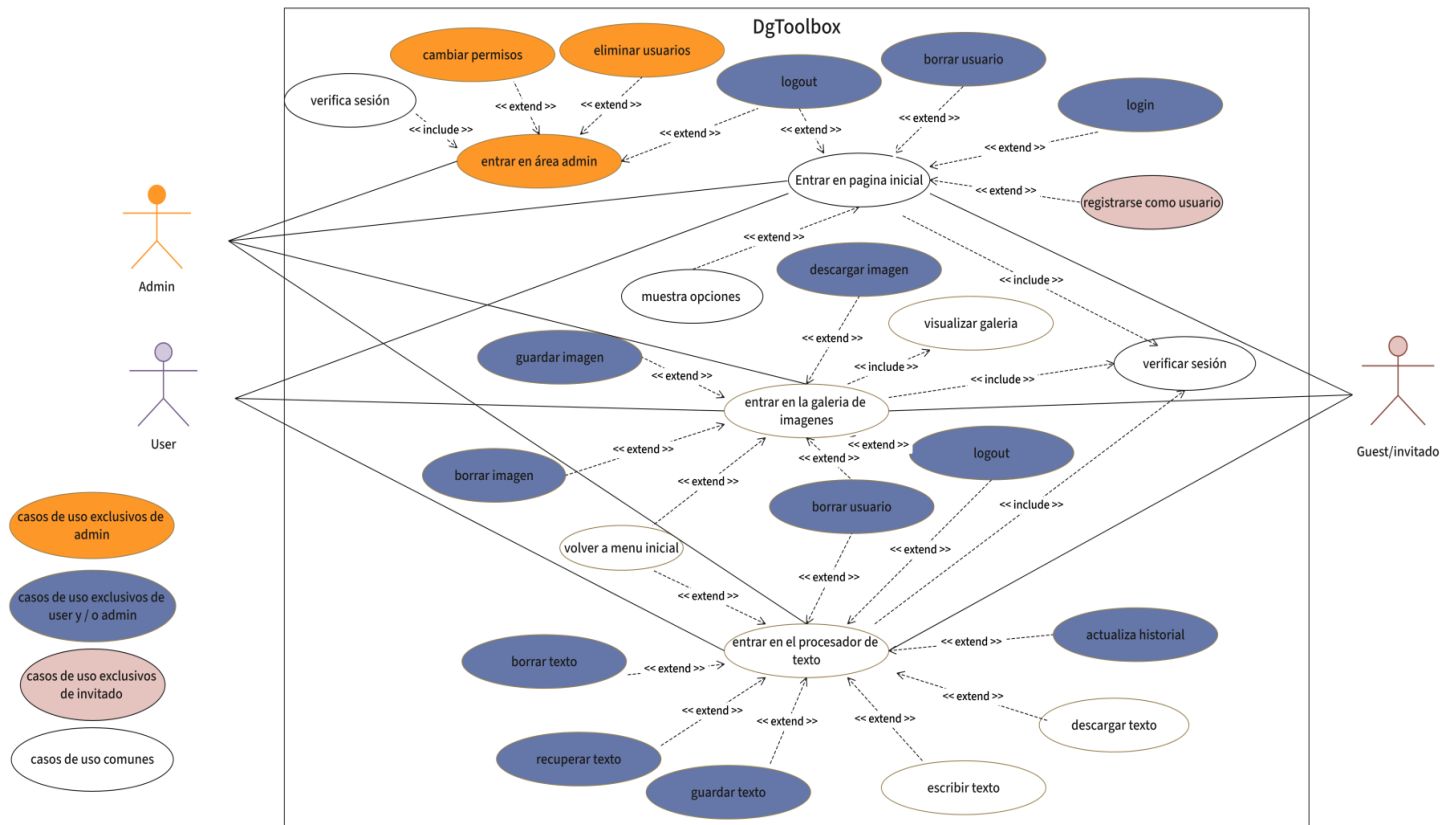


Imagen 6 diagrama de casos de uso (Mocups, 2023)

12.3. Wireframes

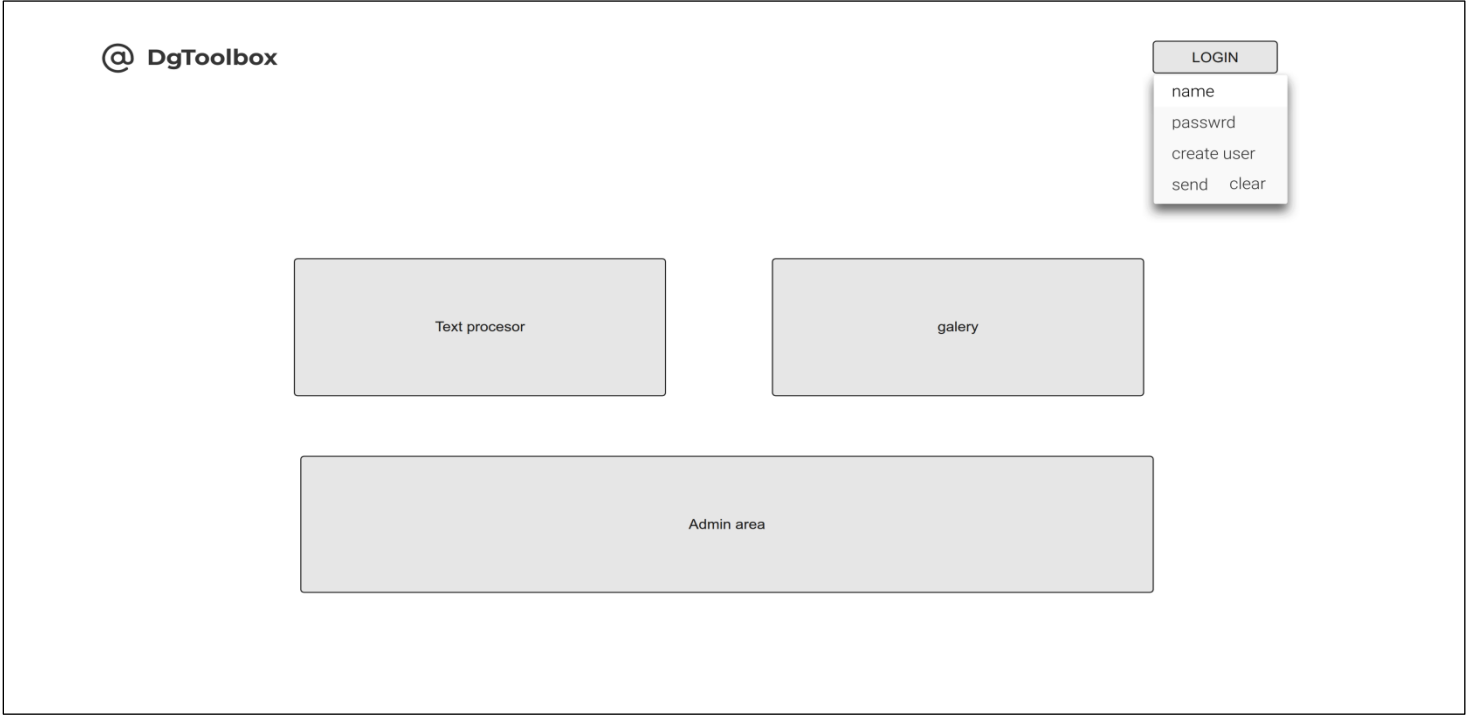


Imagen 7 wireframe menú principal (Mocups, 2023)

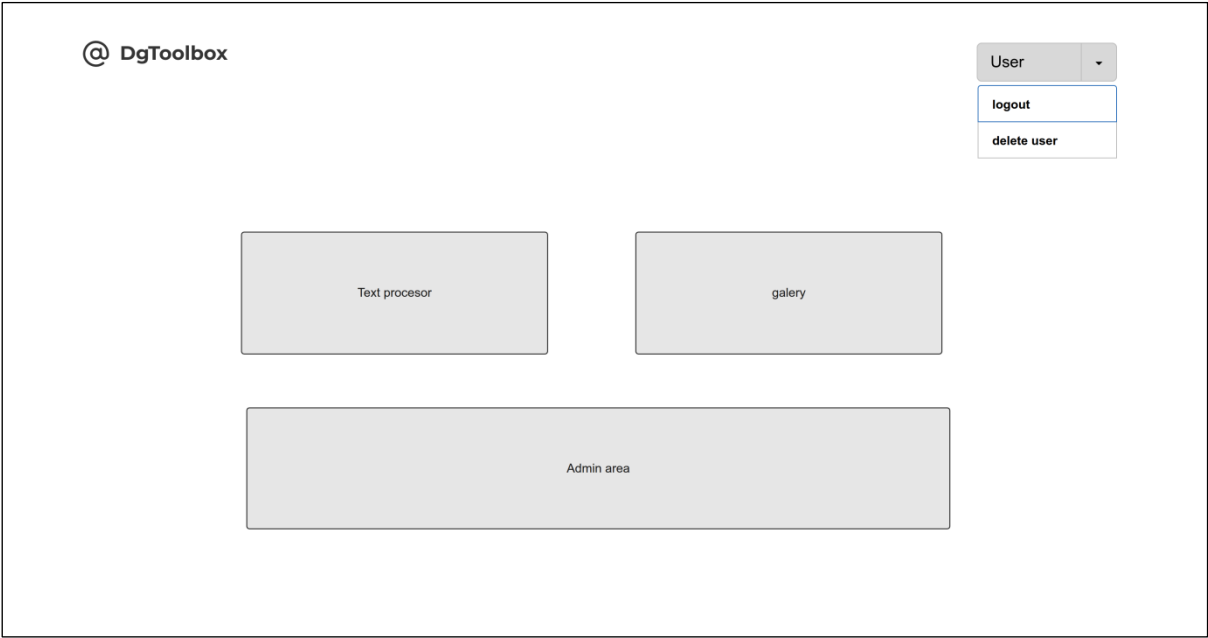


Imagen 8 wireframe menú principal usuario registrado (Mocups, 2023)

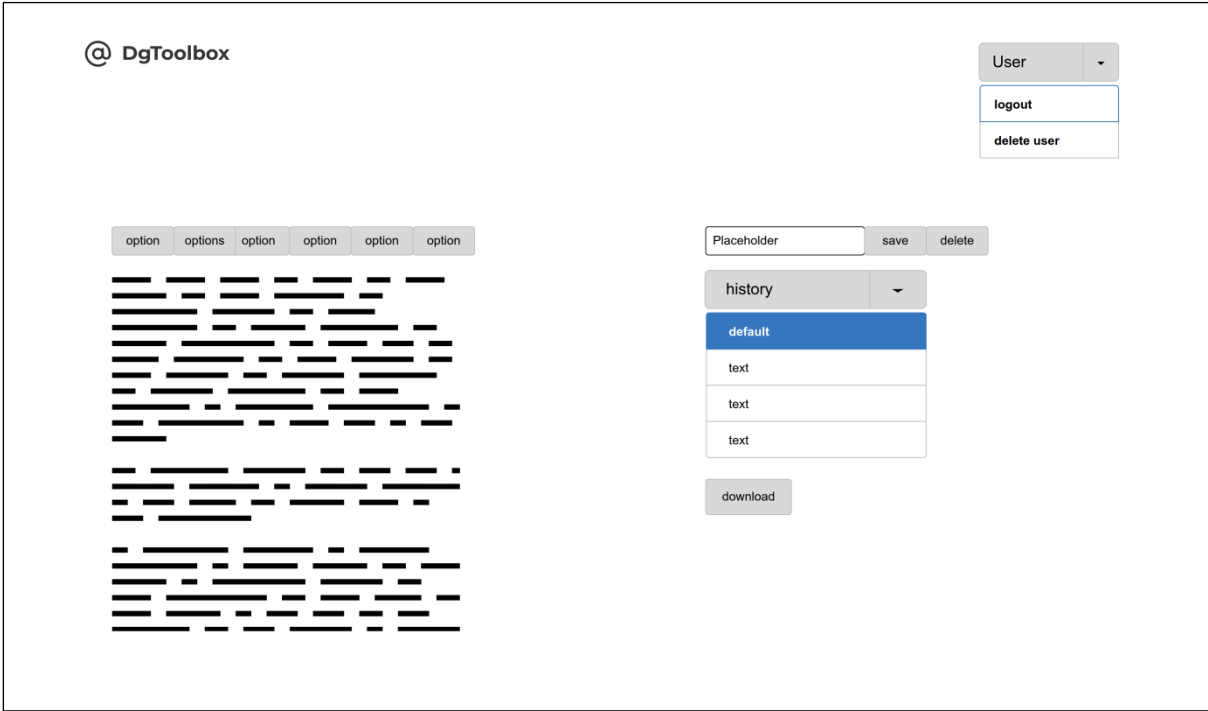


Imagen 9 wireframe procesador de texto (Mocups, 2023)

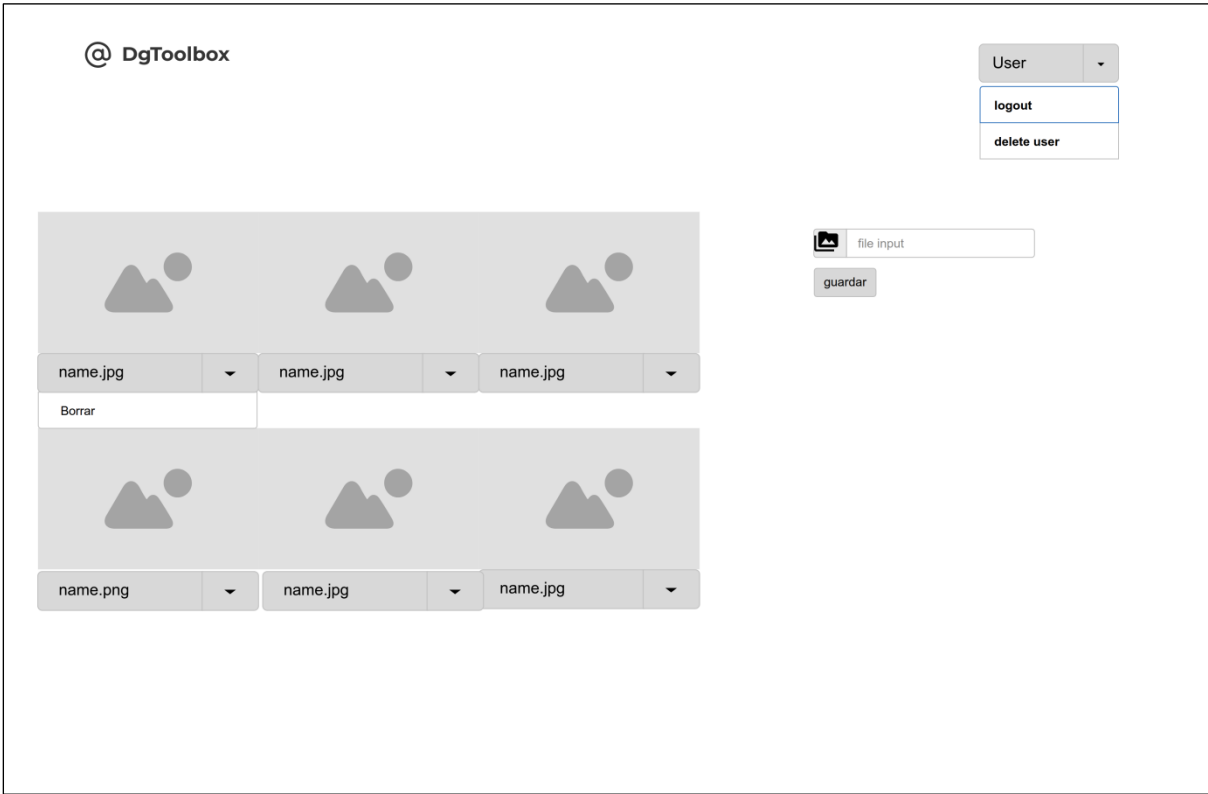


Imagen 10 wireframe galería de imágenes (Mocups, 2023)

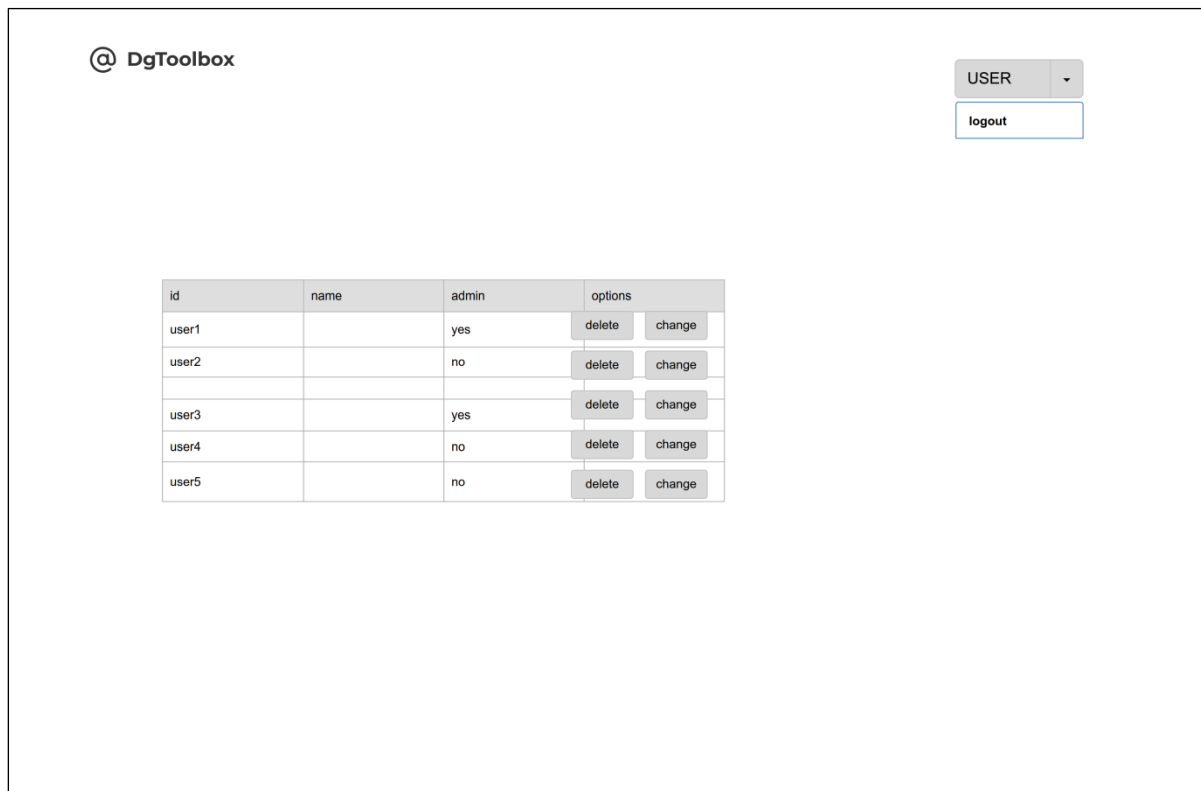


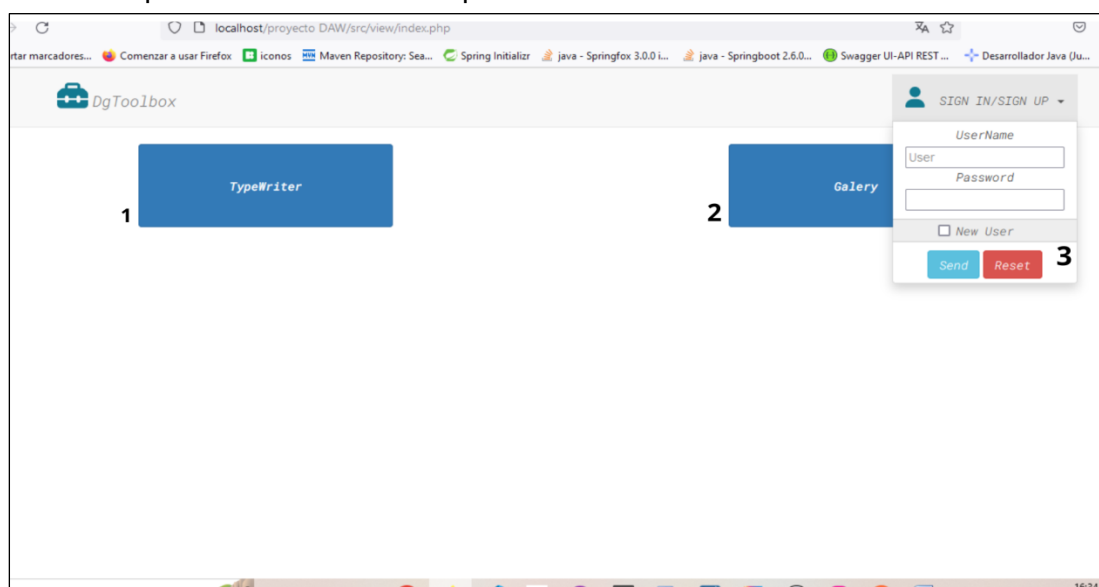
Imagen 11 wireframe área admin (Mocups, 2023)

12.4. Manual de usuario

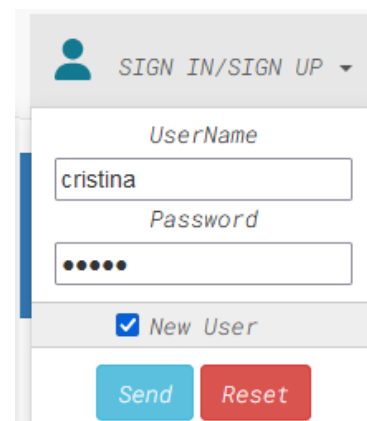
Bienvenido al manual de usuario de DgToolbox, este manual lo dividiremos en varias secciones: Menú principal, Manejo de TypeWriter, Manejo de Galery, manejo de Admin Area.

Menú principal:

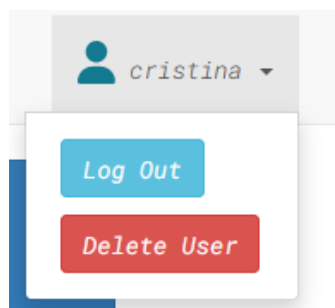
Si entramos en la aplicación este será el primer menú que veamos y en el si no hemos iniciado sesión podremos observar una pantalla como esta.



- 1- Nos conducirá al procesador de texto de la app. Como invitado o con el perfil de usuario con el que nos hayamos registrado.
- 2- Nos conducirá al procesador de texto de la app. Como invitado o con el perfil de usuario con el que nos hayamos registrado. Como invitado no podremos interactuar con ella hasta que nos registremos.
- 3- Menú desplegable para utilizar el sistema de login. En el podemos darnos de alta rellenando los campos UserName, Password y haciendo click en el checkbox de New User. Si no marcamos la opción New User tratara de hacer login con los datos introducidos, dándonos acceso al usuario previamente dado de alta en nuestra aplicación. Si desea reiniciar el formulario solo necesita presionar el botón Reset.

A screenshot of a web form for signing in or signing up. At the top, there is a dropdown menu with a person icon and the text "SIGN IN/SIGN UP". Below this, there are two input fields: "UserName" with the text "cristina" and "Password" with masked characters. A checkbox labeled "New User" is checked. At the bottom, there are two buttons: "Send" (blue) and "Reset" (red).

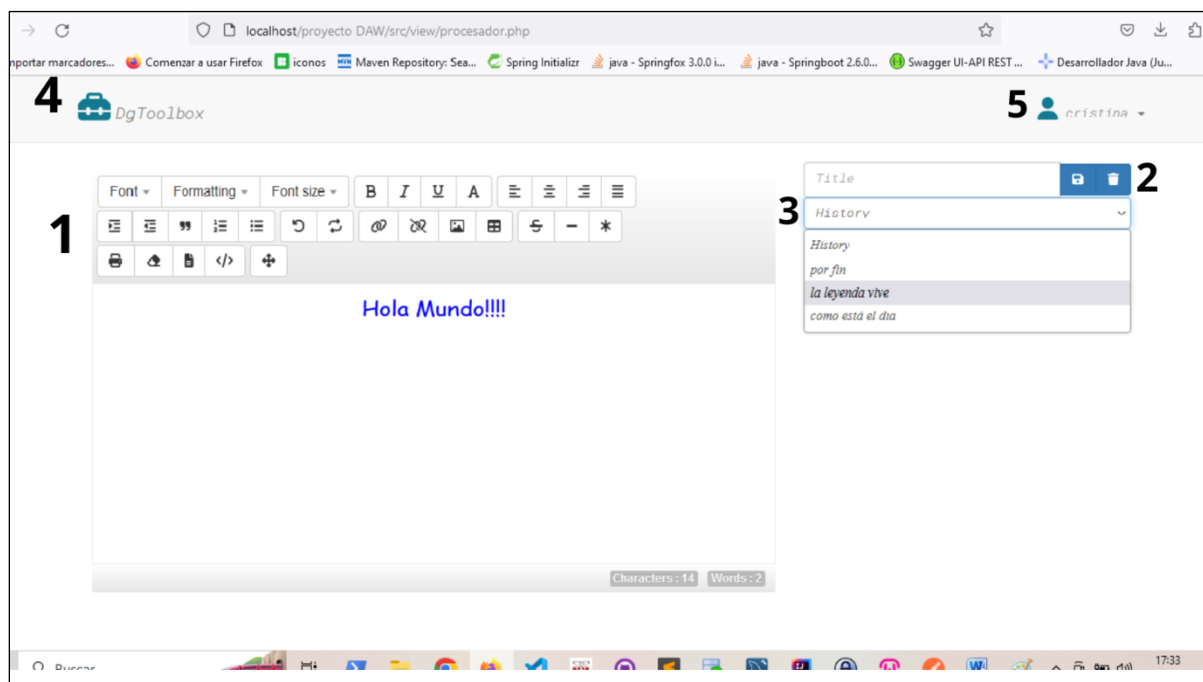
Si pulsas en el cuando has iniciado una sesión podrás ver las opciones de gestion de usuario. Aquí podrás cerrar sesion y borrar el usuario con el que iniciaste sesión.



Una vez nos registremos o iniciemos sesión podremos acceder a la herramienta TypeWriter o Galery. Recuerda que sin iniciar sesión solo podrás interactuar con el TypeWriter.

TypeWriter:

Pasaremos a explicar cómo usar la herramienta TypeWriter como usuario, al final de esta sección puede encontrar las características habilitadas para un usuario invitado (Guest).



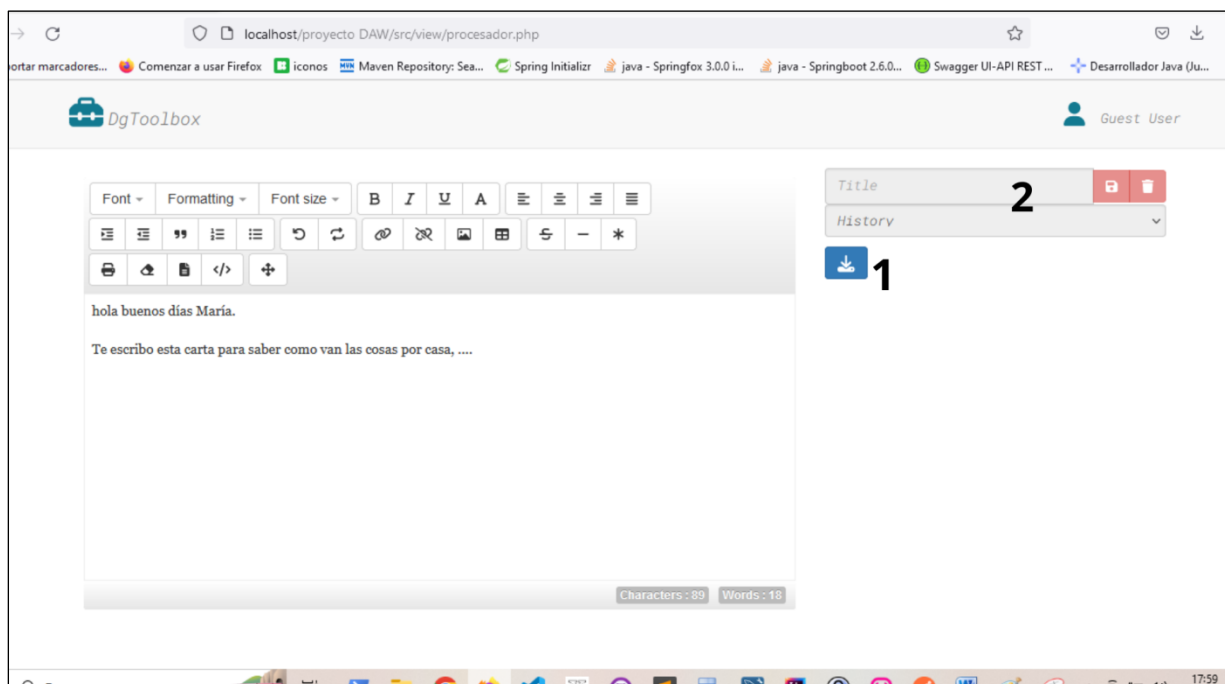
- 1- Área de escritura, aquí escribirás y editarás los textos, tiene multitud de funciones desde las básicas de formato hasta introducir imágenes, tablas, sangrías, vínculos externos, modo de pantalla completa, y muchos más.
- 2- Aquí deberás rellenar el cuadro de texto para indicar el nombre con el que quieres que quede guardado el archivo, o rellenarlo con el nombre del archivo que quieres borrar.

Nota: al cargar un texto desde el historial se rellenará solo.

- 3- Este menú desplegable corresponde al historial de textos persistidos con tu usuario, si aun no guardaste un texto aparecerá la entrada por defecto History. De lo contrario podrás seleccionar uno de tus archivos y cargarlo inmediatamente al área de texto para modificarlo o leerlo.
- 4- Icono DgToolbox pulsa aquí para volver al menú principal.
- 5- Pulsa en este menú desplegable para ver las opciones de gestión de usuario. Como vimos en el punto 3 de la sección "menú principal".

Nota: El botón hacia el enlace de descarga de textos aparecerá cuando cargues un texto usando el historial antes de descargar un texto siendo usuario deberás persistirlo en la base de datos. Puedes observar el botón en la captura de pantalla que se mostrará a continuación en la explicación para el rol de invitado.

Si accediera a TypeWriter como invitado la interfaz aparecera de la siguiente manera.



- 1- Botón hacia el enlace de descarga, este boton te permitira acceder a un enlace de descarga donde descargar el texto escrito aparecera al pulsar una tecla en el area de texto del procesador.
- 2- No podras manipular ni las opciones de guardar y borrar textos ,ni tampoco usar el historial apareceran deshabilitados.

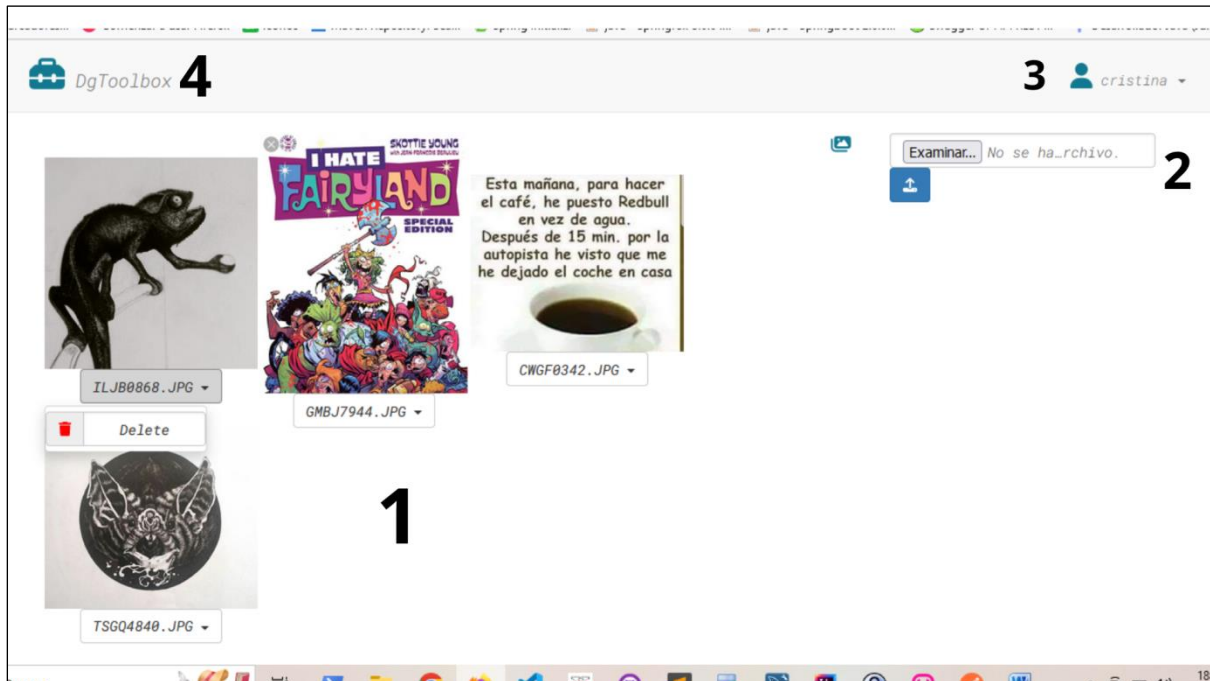
El resto de funciones seran las mismas que las vistas en el apartado de usuario registrado.

Galery:

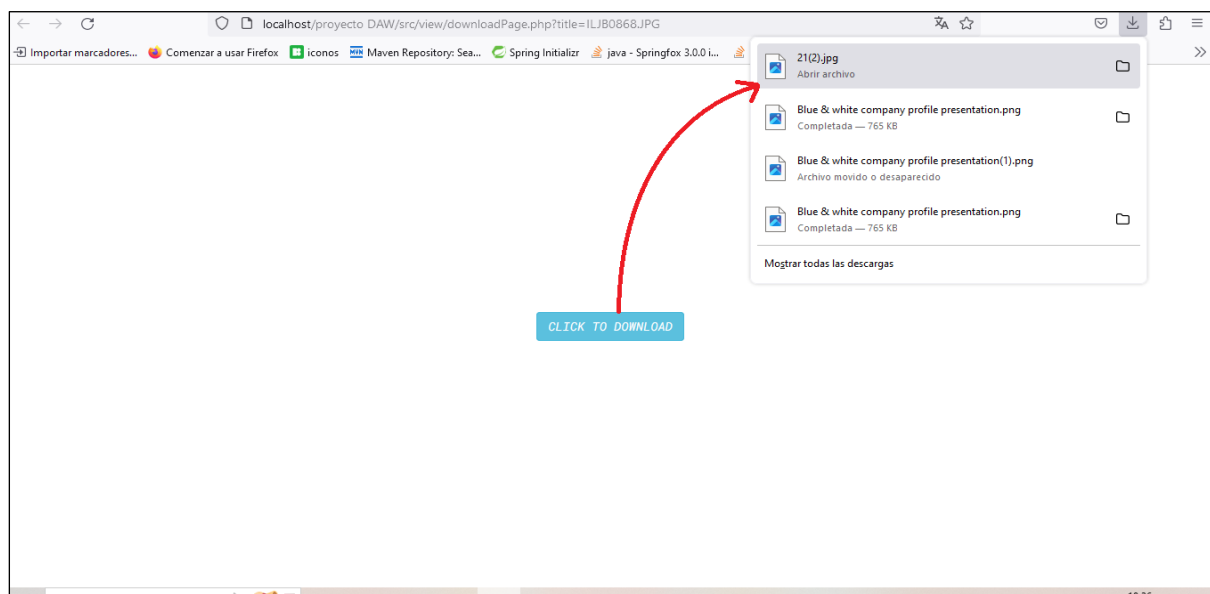
Al entrar en la herramienta galery como usuario registrado encontraremos diferentes elementos como se muestra en la imagen. Estos son:

- 1- Esta zona corresponde al álbum de fotos, si el usuario cosn el que inicies sesion ya tiene persistidas imágenes te apareceran desplegadas en esta área. Tods ellas con un menu desplegable que contiene el botón con el que podrás borrar la imagen en cuestion. Y si deseas descargar la imagen en el equipo al cual estas conectado solo tienes que pulsar en ella, para que seas enviado a la pestaña con el enlace de descarga.

- 2- En este input deberás cargar la imagen que deseas sea persistida asociada a tu cuenta, en la base de datos.
- 3- Como se detalla en el punto 3 de la seccion “menú principal”, este menú desplegable muestra las opciones de gestion de usuario.
- 4- Al pulsar en este icono nos devolvera al menú principal.



La pestaña que te entrega el enlace de descarga de cualquier archivo ya sean imágenes o textos es como la captura que puedes observar a continuacion:

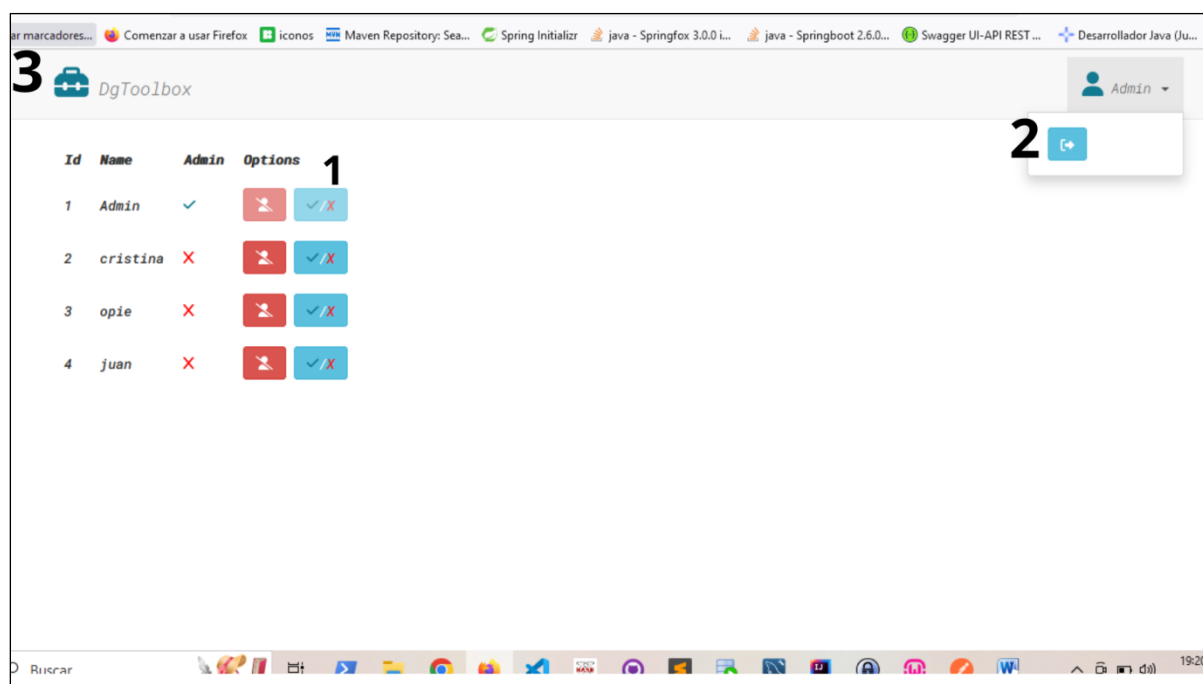


Admin Area:

Esta área solo será visible para usuarios con permisos de administrador. Una vez te hayas logueado como administrador en la aplicación podrás observar la opción de “Admin Area” en el menú principal.



Una vez dentro observaremos la siguiente interfaz:

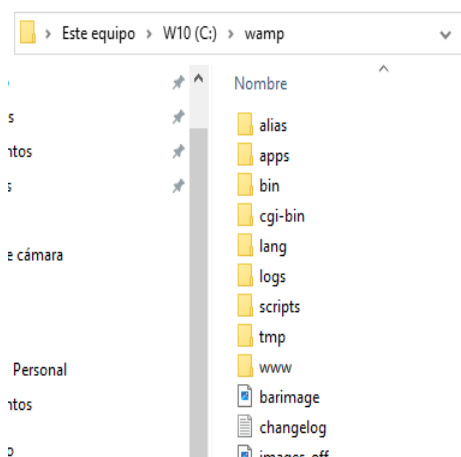


- 1- Esta parte es la tabla de administrador, aquí se podrá visualizar todos los usuarios de la app y sus permisos, gracias a la columna Admin podemos verificar el estado de los permisos, diferenciara entre los usuarios que si son “admin” (check) y cuales no (“X”). Además en la columna “options” podemos utilizar el boton rojo para eliminar al usuario y toos sus archivos asociados, y por otro lado el boton azul para cambiar el estado de los privilegios del usuario de la tabla.
- 2- Este menú desplegable muestra la opcion para desloguearse.
- 3- Al pulsar en el icono nos devolvera al menú principal.

Como podrá observar no permite que pueda borrar su usuario desde la tabla de administrador ni modificar sus privilegios de manera recursiva, tampoco desde ningun menú desplegable de la aplicación esto es para evitar borrados accidentales de usuarios vitales para la gestion de la app y para evitar errores en la coherencia de la aplicación, solo otro admin puede borrar a un usuario con rol de administrador, asegurandose de que siempre quede un admin existente en funcionamiento dentro de la aplicacion.

12.5. Manual de despliegue

Para el despliegue de la aplicación y poder verificar su desarrollo y ejecutar pruebas de funcionamiento y validación utilicé como expliqué en el apartado de tecnologías del proyecto WAMP. Para que el servidor local pueda desplegar el proyecto requiere de un paso muy sencillo y es pasar la carpeta del proyecto a la carpeta “www”, ubicada dentro de la carpeta wamp que se encontrara donde se destinase en la instalación. Una vez depositado el proyecto dentro bastara con escribir en el navegador `http://localhost/{nombre del proyecto}/{recurso}`.



También habrá que importar el script de sql en MyphpAdmin, herramienta de wamp, una vez dentro usaremos el usuario root que por defecto no tiene contraseña, y crearemos una nueva base de datos llamada db_proyecto_daw (como el título del script sql) con la codificación utf8_general_ci, y seguido pulsaremos en la opción **import**, y rellenaremos el input que nos pide el script .sql y pulsaremos en importar abajo del formulario.

The image consists of two screenshots of the phpMyAdmin interface. The top screenshot shows the 'Bases de datos' (Databases) page. On the left sidebar, a tree view shows databases: 'Nueva', 'curso_php', 'db_proyecto_daw', 'information_schema', 'mysql', and 'pac_dwes'. The main panel has a tabbed interface with 'Bases de datos' selected. Below the tabs is a 'Crear base de datos' (Create database) section with a text input for the database name (placeholder: '{nombre bd}'), a dropdown for the character set (selected: 'utf8_general_ci'), and a 'Crear' button. The bottom screenshot shows the 'Importando en la base de datos "{nombre bd}"' (Importing into the database "{nombre bd}") page. The 'Importar' (Import) tab in the top navigation bar is circled in blue. The main panel shows the 'Archivo a importar:' (File to import:) section. It includes instructions about file formats (gzip, bzip2, zip) and a search box for the file. The search box contains 'db_proyecto_daw.sql' and is underlined with a blue line. Below the search box, there is a note about dragging files and a dropdown for the file's character set, which is set to 'utf-8'.

