



UNIVERSIDAD ANDINA DEL CUSCO
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



TESIS

**SISTEMA DE GESTOR DE ARCHIVOS BASADO EN LA
TECNOLOGIA BOOTSTRAP EN EL ÁREA DE ARCHIVOS DE LA
GERENCIA REGIONAL DE ENERGÍA, MINAS E
HIDROCARBUROS - CUSCO**

Línea de Investigación: Desarrollo de Software

Presentado por:

Bach. GALLEGOS YUCRA, LUIS ADRIANO
0009-0000-1441-0850

Para optar el Título Profesional de:
INGENIERO DE SISTEMAS

Asesor:

Mgt. Ing. HOLGUÍN HERRERA, MELISA
BETYS
0000-0001-5347-9252

CUSCO – PERÚ

2023



Metadatos

Datos del autor	
Nombres y apellidos	Nwku'Cf tkepqi'c enqi qu'c wetc
Número de documento de identidad	9475; 599
URL de Orcid	https://orcid.org/0009-0000-1441-0850
Datos del asesor	
Nombres y apellidos	Ing.Melisa Bety Holguín Herrera
Número de documento de identidad	43109012
URL de Orcid	https://orcid.org/0000-0001-5347-9252
Datos del jurado	
Presidente del jurado (jurado 1)	
Nombres y apellidos	Ing.Maria Isabel Acurio Gutierrez
Número de documento de identidad	29715419
Jurado 2	
Nombres y apellidos	Ing.Velia Ardiles Romero
Número de documento de identidad	40845490
Jurado 3	
Nombres y apellidos	Ing.Harry Yeison Gonzales Condori
Número de documento de identidad	24003082
Jurado 4	
Nombres y apellidos	Ing.Félix Enrique Huaman Ataulluco
Número de documento de identidad	25001941
Datos de la investigación	
Línea de investigación de la Escuela Profesional	Desarrollo de Software



SISTEMA DE GESTOR DE ARCHIVOS BASADO EN LA TECNOLOGIA BOOTSTRAP EN EL ÁREA DE ARCHIVOS DE LA GERENCIA REGIONAL DE ENERGÍA, MINAS E HIDROCARBUROS - CUSCO

por LUIS ADRIANO GALLEGOS YUCRA

Fecha de entrega: 21-feb-2024 02:28p.m. (UTC-0500)

Identificador de la entrega: 2300828300

Nombre del archivo: Tesis_2023_GALLEGOS_YUCRA_LUIS_ADRIANO_COMPLETO.pdf (4.36M)

Total de palabras: 24007

Total de caracteres: 138815

MGT.ING.MELISA BETYS HOLGUIN HERRERA
Docente de la E.P. de Ingeniería de Sistemas - UAC



UNIVERSIDAD ANDINA DEL CUSCO ●
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



TESIS

SISTEMA DE GESTOR DE ARCHIVOS BASADO EN LA
TECNOLOGIA BOOTSTRAP EN EL ÁREA DE ARCHIVOS DE LA
GERENCIA REGIONAL DE ENERGÍA, MINAS E
HIDROCARBUROS - CUSCO

Línea de Investigación: Desarrollo de Software

Presentado por:

Bach. GALLEGOS YUCRA, LUIS ADRIANO

Para optar el Título Profesional de:

INGENIERO DE SISTEMAS

Asesor:

Mgt. Ing. HOLGUÍN HERRERA, MELISA
BETYS

CUSCO – PERÚ

2023

MGT. ING. MELISA BETYS HOLGUIN HERRERA
Docente de la E.P. de Ingeniería de Sistemas - UAC



TECNOLOGIA BOOTSTRAP EN EL ÁREA DE ARCHIVOS DE LA GERENCIA REGIONAL DE ENERGÍA, MINAS E HIDROCARBUROS - CUSCO

INFORME DE ORIGINALIDAD

2%

EN

INDICE DE SIMILITUD

1%

FUENTES DE INTERNET

0%

PUBLICACIONES

0%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	warbletoncouncil.org Fuente de Internet	1%
2	Submitted to Wilfrid Laurier University Trabajo del estudiante	<1%
3	Submitted to Tshwane University of Technology Trabajo del estudiante	<1%
4	magazine.logigear.com Fuente de Internet	<1%
5	www.coursehero.com Fuente de Internet	<1%
6	repository.futminna.edu.ng:8080 Fuente de Internet	<1%
7	Create Web Charts With jqPlot, 2014. Publicación	<1%

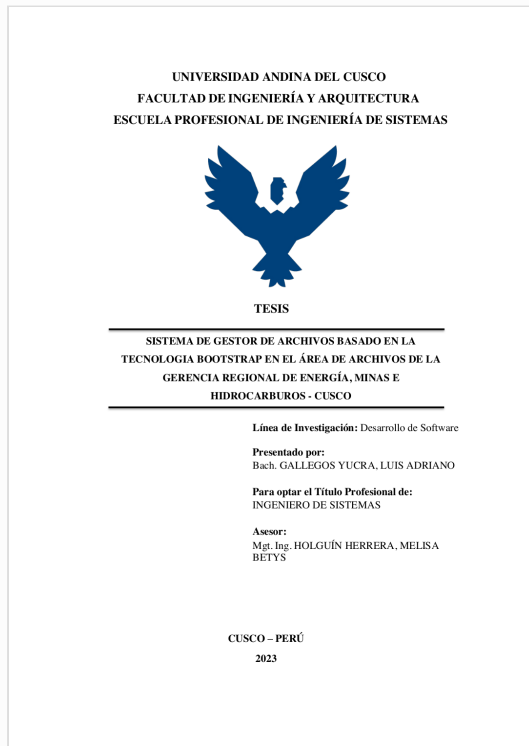


Recibo digital

Este recibo confirma que su trabajo ha sido recibido por **Turnitin**. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega: LUIS ADRIANO GALLEGOS YUCRA
Título del ejercicio: Tesis sin deposito
Título de la entrega: SISTEMA DE GESTOR DE ARCHIVOS BASADO EN LA TECNOLO...
Nombre del archivo: Tesis_2023_GALLEGOS_YUCRA_LUIS_ADRIANO_COMPLETO.pdf
Tamaño del archivo: 4.36M
Total páginas: 122
Total de palabras: 24,007
Total de caracteres: 138,815
Fecha de entrega: 21-feb.-2024 02:28p. m. (UTC-0500)
Identificador de la entre... 2300828300



MGT.ING.MELISA BETYS HOLGUIN HERRERA
Docente de la E.P. de Ingeniería de Sistemas - UAC

Derechos de autor 2024 Turnitin. Todos los derechos reservados.



Dedicatoria

En este importante hito de mi vida académica, quiero dedicar esta tesis a cada uno de ustedes, que han sido parte fundamental de mi camino hacia el éxito. Con su, apoyo y amor incondicional han sido pilares que me han sostenido en los momentos de dificultad y han celebrado mis triunfos con alegría.

A ti, querida mamá Ana María, agradezco por tu amor infinito, por tu paciencia y por ser mi ejemplo de perseverancia. Tus palabras de aliento y tu constante estímulo han sido un impulso en mi camino. Esta tesis es también un tributo a tu dedicación y sacrificio por mi educación.

A ti, querido papá José Luis, agradezco por tu guía y sabiduría. Tu fuerza y determinación me han inspirado a superar obstáculos y a alcanzar mis metas. Tu apoyo incondicional ha sido un regalo invaluable en mi vida.

A ti, querida madrina Armanda, y a ti, querido padrino Ives, agradezco por su cariño y por estar siempre presentes en mi vida. Su apoyo me ha motivado a dar lo mejor de mí mismo en este camino académico.

A mi amada familia, agradezco por el amor que me han brindado. Su presencia y ánimo han sido una fuente de alegría en cada paso que he dado. Esta tesis es también un tributo a la unión y a los valores que me han inculcado.

A mis amigos y amigas, quienes han sido mi refugio en los momentos de estrés y han compartido conmigo risas, lágrimas y experiencias inolvidables, les agradezco por su amistad. Su presencia en mi vida ha enriquecido mi camino y me ha recordado la importancia de mantener el equilibrio entre el trabajo y el disfrute.

A mis adoradas mascotas, Suave y Oreo, quienes han sido mis compañeros fieles y han llenado mi vida de amor y alegría. Gracias por estar siempre a mi lado, brindándome consuelo y ternura. Esta tesis también es un homenaje a su lealtad y amor.

Finalmente, quiero dedicar esta tesis a mí mismo, a mi determinación y perseverancia. A través de altibajos, desafíos y momentos de duda, nunca dejé de creer en mí mismo y en mi capacidad para alcanzar mis metas. Esta tesis es el testimonio de mi arduo trabajo, dedicación y pasión por mi campo de estudio.



Agradecimientos

En primer lugar, agradezco a Dios por ser mi guía constante y mi fortaleza en cada paso de este camino.

A mis queridos padres, quienes han sido mi mayor fuente de inspiración y apoyo incondicional. Gracias por creer en mí y por alentarme en cada paso de mi trayectoria académica. Su amor, paciencia y sacrificio han sido fundamentales en mi formación y en la consecución de este logro.

A la Universidad Andina del Cusco, a la Facultad de Ingeniería y Arquitectura y a mi Escuela Profesional de Ingeniería de Sistemas, les agradezco por brindarme la oportunidad de formarme en un ambiente académico de calidad. Gracias por proporcionarme los recursos y las herramientas necesarias para adquirir conocimientos y habilidades en mi campo de estudio.

A mi asesora, la Ing. Holguín Herrera, Melisa Betys, quiero expresar mi profundo agradecimiento. Gracias por su guía, orientación y apoyo a lo largo de esta tesis. Sus conocimientos y experiencia han sido invaluable para el desarrollo de mi investigación.

Agradezco también a todos mis profesores y profesoras, cuya dedicación y pasión por la enseñanza han dejado una huella imborrable en mi formación académica. Gracias por compartir su conocimiento, por desafiarme intelectualmente y por motivarme a alcanzar mis metas.



Índice General

Índice General.....	III
Índice de Tablas.....	V
Índice de Figuras	VI
Introducción.....	1
Abstract.....	2
CAPÍTULO 1 - Problema de Investigación.....	3
1.1 Ámbito de influencia.....	3
1.1.1 Ámbito de influencia teórica.....	3
1.1.2 Área de dominio.....	3
1.1.3 Línea de investigación	3
1.2 Planteamiento del problema.....	3
1.2.1 Descripción de la situación actual del lugar de intervención.....	3
1.2.2 Descripción del problema	8
1.2.3 Formulación del problema	12
1.2.4 Objetivos	12
Objetivo General.....	12
Objetivos Específicos	12
1.2.5 Justificación	13
1.2.6 Alcances y limitaciones	13
CAPÍTULO 2 – Marco Teórico.....	14
2.1 Antecedentes del desarrollo, implementación o transferencia tecnología.....	14
2.1.1 Antecedentes Nacionales	14
2.1.2 Antecedentes Internacionales.....	16
2.2 Bases teórico – científicos.....	19
CAPÍTULO 3 – Desarrollo, Implementación o Transferencia Tecnológica.....	29
3.1.2.1 Fase 1 Planificación	35



3.1.2.2 Fase 2 Diseño	39
3.1.2.3 Fase 3 Codificación	44
3.1.2.4 Fase 4 Pruebas	91
3.1.2.5 Fase 5 Integración continua:	93
CAPÍTULO 4 – Resultados	96
4.1 Comprobación de la prospectiva.....	96
4.2 Cumplimiento de objetivos	99
4.3 Contribuciones (impacto)	100
Glosario	102
Conclusiones.....	103
Recomendaciones	104
Referencias	105
Anexos.....	108



Índice de Tablas

Tabla 1 Los pasos principales de la metodología XP en el proyecto.	32
Tabla 2 Caso I de historia de usuario: Buscar archivos por nombre	36
Tabla 3 Caso II de historia de usuario: Subir archivos al sistema	36
Tabla 4 Caso III de historia de usuario: Organizar archivos en carpetas	36
Tabla 5 Caso IV de historia de usuario: Descargar archivos del sistema.....	37
Tabla 6 Caso de prueba 1: Subir un archivo a una categoría específica en el sistema de gestor de archivos.	37
Tabla 7 Caso de prueba 2: Eliminar un archivo de una categoría específica en el sistema de gestor de archivos.	38
Tabla 8 Caso de prueba 3: Descargar un archivo desde una categoría específica en el sistema de gestor de archivos.	38
Tabla 9 Caso de prueba 4: Visualizar los archivos en una categoría específica en el sistema de gestor de archivos.	39
Tabla 10 Información cada carpeta de la estructura del gestor de archivos	45
Tabla 11 Aportes de los Antecedentes	98
Tabla 12 Cumplimiento de objetivos.....	99
Tabla 13 Presupuesto para el proyecto de investigación desarrollo del sistema de gestor de archivos.....	109
Tabla 14 Matriz de consistencia del sistema gestor de archivos para la Gremh en el área de Archivos.....	110



Índice de Figuras

Figura 1 Organigrama de la Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco)	4
Figura 2 Logo de la Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco).....	6
Figura 3 Vista Frontal de la Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos -Cusco)	7
Figura 4 Vista Interna de la Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos -Cusco)	7
Figura 5 Mapa de la localización de la Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco)	8
Figura 6 Ficheros Color Azul de Gerencia General	9
Figura 7 Ficheros Color Verde de Informes de Fiscalización	9
Figura 8 Ficheros Color Blanco de área de Hidrocarburos	10
Figura 9 Módulo de archivos actual de la Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco).....	11
Figura 10 Fases de la metodología XP (Programación Extrema)	31
Figura 11 Detalle de actividades Plan de trabajo de desarrollo de gestor de archivos en la Gremh (gerencia regional de energía, minas e hidrocarburos - Cusco) – Área de Documentos (https://goo.su/qQp6PGL)	44
Figura 12 Diagrama de caso de uso del sistema de gestor de archivos para la Gremh ..	35
Figura 13 Diagrama de componentes UML de la arquitectura del sistema de gestor de archivos.....	40
Figura 14 Parte I maquetación (Login de usuario) de proyecto gestor de archivos Gremh.	41



Figura 15 Parte II maquetación (Inicio de usuario) de proyecto gestor de archivos Gremh.	41
Figura 16 Parte III maquetación (Modulo Categoría) de proyecto gestor de archivos Gremh.	42
Figura 17 Parte IV maquetación (Modulo Archivo) de proyecto gestor de archivos Gremh.	43
Figura 18 Iniciar el servidor web XAMPP conjunto el Apache y al MySQL.....	44
Figura 19 Estructura de carpeta para el desarrollo del gestor de archivos	45
Figura 20 Categorías.php, métodos para agregar, eliminar, obtener y actualizar categorías en una base de datos SQL.....	46
Figura 21 Establecer una conexión con una base de datos utilizando la extensión MySQL	47
Figura 22 Operaciones relacionadas con el manejo de archivos en la base de datos	48
Figura 23 Métodos para agregar usuarios, buscar usuarios repetidos y realizar el inicio de sesión de usuarios en la base de datos.	50
Figura 24 Formato y diseño para la formulación y animación de la página HTML	52
Figura 25 Archivo en formato de imágenes.	53
Figura 26 Funciones para agregar, eliminar, obtener y actualizar categorías mediante el uso de solicitudes AJAX.	53
Figura 27 Funciones para administrar archivos mediante solicitudes AJAX.....	56
Figura 28 Librerías para el desarrollo del gestor de archivos	57
Figura 29 Clase "Categorías" para actualizar, agregar, eliminar y obtener una categoría en la base de datos.	59
Figura 30 Eliminar un archivo del servidor y eliminar el registro en la base de datos ..	61



Figura 31 Subir Múltiples archivos al servidor y registrarlos	62
Figura 32 Ruta de archivo en el servidor.....	63
Figura 33 Proceso de inicio de sesión de usuario.....	64
Figura 34 Agrega un nuevo usuario utilizando la clase "Usuario"	65
Figura 35 Cierra la sesión del usuario y lo redirecciona al inicio.	65
Figura 36 Muestra las categorías asociadas a un usuario específico.....	66
Figura 37 Tabla HTML con las categorías asociadas al usuario específico.....	67
Figura 38 Funcionalidades para descargar, visualizar y eliminar archivos que almacena el usuario.	69
Figura 39 Página de administración de categorías (agregar, actualizar y eliminar categorías).....	71
Figura 40 Scripts de librerías en un archivo HTML.....	72
Figura 41 Interfaz del gestor de archivos y proporciona funcionalidad para agregar y visualizar archivos.	73
Figura 42 Estructura de una página web y establece una barra de navegación con enlaces a las diferentes secciones del sitio	74
Figura 43 Verifica la sesión de usuario y muestra una página de bienvenida si el usuario ha iniciado sesión	75
Figura 44 Deshabilitar la opción de ver el listado de archivos.....	76
Figura 45 Página de inicio de sesión	76
Figura 46 Página de registro de usuario con un formulario	77
Figura 47 Inicio de MySQL Workbench y crear la base de datos con el nombre de local adrián	78
Figura 48 Creación de la base de datos llamado “gestorgremh”	79



Figura 49 Creación de la tabla “t_usuarios”	79
Figura 50 Creación de la tabla “t_categorias”	80
Figura 51 Llave foránea "fkCategoriaUsuario"	80
Figura 52 Diagrama de clase de t_Categorias con t_usuarios	80
Figura 53 Creación de la tabla “t_archivos”	81
Figura 54 Agregar la llave foránea con la tabla “t_Categorias”	82
Figura 55 Diagrama de clases incluyendo la relación de las tablas	82
Figura 56 Maquetación del Login	83
Figura 57 Crear el Registro	83
Figura 58 Desarrollo del Menú horizontal	84
Figura 59 CRUD de archivos	84
Figura 60 Programación para registrar usuario	84
Figura 61 Verificación del nuevo usuario registrado en la base de datos “t_usuarios”	85
Figura 62 Programación Login de usuario	85
Figura 63 Implementación del Módulo Categorías	86
Figura 64 Verificar el registro de la nueva categoría en la base de datos	87
Figura 65 Módulo de Eliminar Categoría.....	87
Figura 66 Módulo de Editar Categoría.....	88
Figura 67 Guardar archivos en el módulo Archivo	88
Figura 68 Verificación de los archivos en la base de datos y en la estructura del proyecto.	89
Figura 69 Funcionalidad de Descargar, Visualizar e Eliminar los archivos	89



Figura 70 Descargar Archivos.....	90
Figura 71 Visualizar Archivos.....	90
Figura 72 Eliminar Archivos	91
Figura 73 Local del área de archivos de la oficina de administración	93
Figura 74 Manual de uso del sistema de Gestor de Archivos Gremh (https://goo.su/UC1MbOX).....	94
Figura 75 Área de Minería y Concesión minera	94
Figura 76 Área de cobertura energética y sistema de ventanilla única para formalización minera	95
Figura 77 Área de Proyectos	95



Introducción

La gestión de la información es un aspecto crítico en cualquier organización, ya que la eficiente administración de los datos y documentos se traduce en una mejor toma de decisiones y, por ende, en una mejora de la productividad y eficiencia en la gestión general de la organización. La Gerencia Regional de Energía, Minas e Hidrocarburos de Cusco es una entidad que maneja grandes volúmenes de información relacionados con la exploración, extracción y producción de energía, minerales e hidrocarburos, lo que hace que la gestión de los archivos sea una tarea compleja y crítica.

Es en este contexto en el que se propone la implementación de un sistema de gestor de archivos basado en la tecnología Bootstrap que es un framework de diseño web que permite crear interfaces responsivas y adaptables a diferentes dispositivos, lo que facilita el acceso a los archivos. La implementación de un sistema de gestor de archivos basado en Bootstrap permitirá al área de archivos de la Gerencia Regional de Energía, Minas e Hidrocarburos de Cusco tener una gestión más eficiente y segura de la información, lo que se traducirá en una toma de decisiones más informada y en una mejora en la productividad de la organización. El propósito de este proyecto es mejorar la gestión de la información en el área de archivos, a través de la implementación de un sistema de gestor de archivos basado en la tecnología Bootstrap. Diseñar e implementar una interfaz de usuario amigable y responsiva que permita el fácil acceso y consulta de los archivos por parte de los usuarios autorizados además de desarrollar una estructura de archivo coherente y fácil de navegar, que permita una rápida localización de la información, también capacitar al personal encargado de la gestión de los archivos en el uso del sistema de gestor de archivos, lo que permitirá una adecuada utilización del sistema y una mejor gestión de la información. En conclusión, la implementación de un sistema de gestor de archivos basado en la tecnología Bootstrap en el área de archivos de la Gerencia Regional de Energía, Minas e Hidrocarburos de Cusco es una medida necesaria y efectiva para mejorar la gestión de la información y, por ende, para mejorar la eficiencia y productividad de la gerencia.



Abstract

Information management is a critical aspect in any organization, as the efficient administration of data and documents translates into better decision-making and, consequently, an improvement in overall organizational productivity and efficiency. The Regional Management of Energy, Mines, and Hydrocarbons in Cusco is an entity that handles large volumes of information related to the exploration, extraction, and production of energy, minerals, and hydrocarbons, making file management a complex and critical task.

It is in this context that the implementation of a file management system based on Bootstrap technology is proposed. Bootstrap is a web design framework that allows for the creation of responsive interfaces adaptable to different devices, facilitating access to files. The implementation of a file management system based on Bootstrap will enable the archives department of the Regional Management of Energy, Mines, and Hydrocarbons in Cusco to have a more efficient and secure information management, leading to more informed decision-making and improved organizational productivity. The purpose of this project is to enhance information management in the archives department through the implementation of a file management system based on Bootstrap technology. This involves designing and implementing a user-friendly and responsive interface that allows easy access and retrieval of files by authorized users. Additionally, it includes developing a coherent and easy-to-navigate file structure, enabling quick information location. The project also aims to train the personnel responsible for file management in using the file management system, ensuring proper utilization and improved information management. In conclusion, the implementation of a file management system based on Bootstrap technology in the archives department of the Regional Management of Energy, Mines, and Hydrocarbons in Cusco is a necessary and effective measure to enhance information management and, consequently, improve the efficiency and productivity of the management.



CAPÍTULO 1 - Problema de Investigación

1.1 Ámbito de influencia

1.1.1 Ámbito de influencia teórica

Enfocarse en el diseño y desarrollo de un sistema informático que permita la gestión eficiente de archivos en la Gerencia Regional de Energía, Minas e Hidrocarburos de Cusco, el cual abarca varias áreas del conocimiento, como la gestión de documentos, la gestión de la información, la tecnología de la información, la seguridad de la información, la organización y estructuración de archivo.

1.1.2 Área de dominio

Tecnologías de Información

Utiliza la tecnología de información de varias formas para mejorar la gestión de información en la organización, algunas de las formas en la que se aplica la tecnología de información en este sistema son: digitalización, almacenamiento, organización, control de versiones y seguridad de los documentos, mejorando así la eficiencia y efectividad de la gestión documental en la organización.

1.1.3 Línea de investigación

Desarrollo de Software

El desarrollo de software desempeña un papel fundamental en el sistema de gestor de archivos, ya que permite crear un sistema informático personalizado que se adapta a las necesidades y requerimientos específicos de la gerencia. Para lograr esto, se utilizará la metodología de desarrollo XP, la cual se destaca por su enfoque ágil y adaptable.

1.2 Planteamiento del problema

1.2.1 Descripción de la situación actual del lugar de intervención.

Descripción general

La GREMH (Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco), es un organismo encargado de supervisar, regular y promover el desarrollo del sector energético, minero y de hidrocarburos a nivel regional. Esta entidad se encarga de implementar y ejecutar políticas nacionales y regionales en estas áreas, así como de fomentar la inversión privada en estos sectores. Además, es responsable de garantizar el cumplimiento de las normativas y regulaciones pertinentes en la exploración y comercialización de recursos energéticos, mineros y de hidrocarburos en su jurisdicción.



El área de archivos, administrada en conjunto con las áreas de administración e informática, se centra principalmente en el almacenamiento físico de documentos y no cuenta con tecnología en ese aspecto. No obstante, la Gremh cuenta con un área de informática.

Descripción del área de informática:

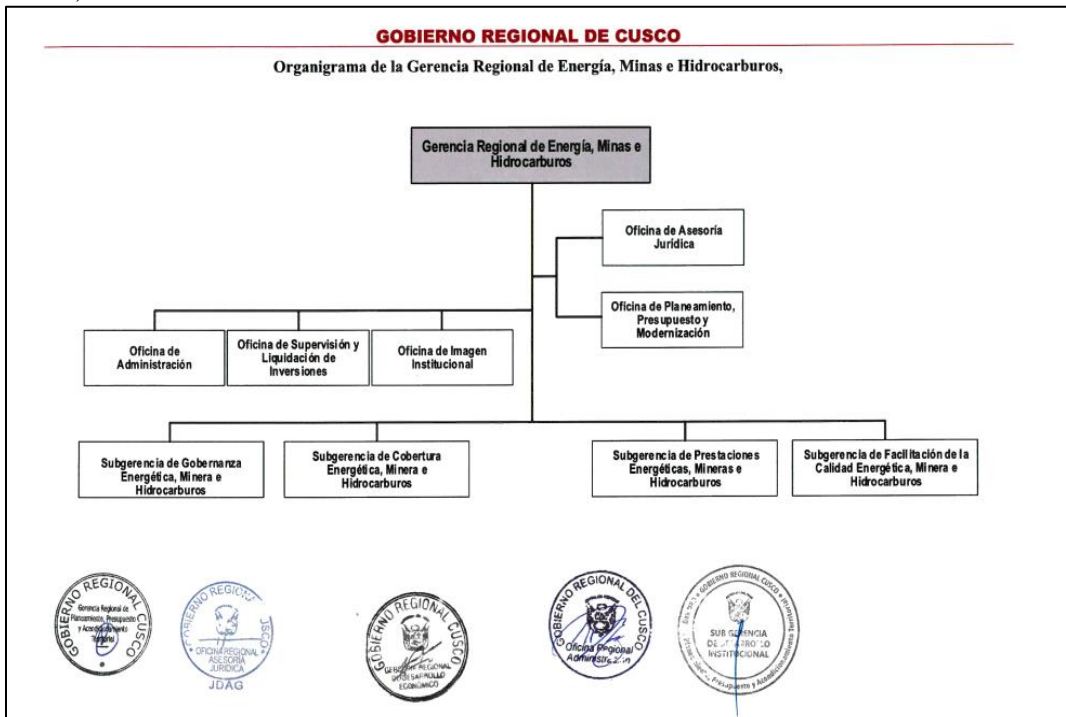
El área de informática, supervisada por un personal especializado, cuenta con un gabinete de redes. Dispone de 2 computadoras y 2 laptops conectadas por red local (cable y Wi-Fi) con Windows 10, procesadores i5 Core y switches administrables. Destacando, la implementación del sistema de gestor de archivos se realizará en la computadora del personal del área, aprovechando los recursos disponibles.

Características Detalladas de los Ordenadores:

Las Computadoras 1 y 2 comparten especificaciones similares, incluyendo Windows 10, procesador Intel Core i5 y 32 GB de RAM, con refrigeración por aire y conectividad cableada y Wi-Fi. Las Laptops 1 y 2 también comparten características idénticas, con sistema operativo Windows 10, procesador Intel Core i5, 16 GB de RAM, refrigeración pasiva y conexión Wi-Fi.

Figura 1

Organigrama de la Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco)



Nota. Organigrama Gremh, por Sitio web Gob.pe– Cusco (<https://goo.su/f9pKtY>)



Procesos principales del área de archivos dentro de la Gremh

La Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco) tiene la responsabilidad de gestionar y supervisar los recursos energéticos, minerales e hidrocarburos en una región determinada. Dentro de la Gremh, el área de archivos se encarga de la gestión eficiente y organizada de los documentos y registros relacionados con las actividades y funciones de la gerencia. Las principales funciones administrativas desarrolladas en este módulo, conforme al organigrama, comprenden:

- 1. Gerencia Regional de Energía, Minas e Hidrocarburos:** Se encarga de la gestión y supervisión de los recursos energéticos, minerales y de hidrocarburos en una región determinada.
- 2. Oficina de Asesoría Jurídica:** Se encarga de proporcionar asesoramiento legal y apoyo a la Gremh. Su función es garantizar que la gerencia cumpla con las leyes.
- 3. Oficina de Planeamiento Presupuesto y Modernización:** Se encarga de planificar y gestionar los recursos financieros y humanos de la Gremh
- 4. Oficina de Administración:** Encargada de gestionar los recursos y servicios necesarios para el funcionamiento de la Gremh, incluyendo la administración del área de archivos de la misma.
- 5. Oficina de Supervisión y Liquidación de Inversiones:** Se encarga de supervisar y evaluar el desempeño de las inversiones realizadas por la Gremh
- 6. Oficina de Imagen Institucional:** Se encarga de gestionar la imagen y la comunicación de la Gremh.
- 7. Subgerencia de Gobernanza Energética, Minas e Hidrocarburos:** Se encarga de promover la gobernanza y la gestión responsable de los recursos energéticos, minerales y de hidrocarburos en la Gremh.
- 8. Subgerencia de Cobertura Energética, Minera e Hidrocarburos:** Se encarga de garantizar la cobertura de la demanda energética, minera e hidrocarburos de la Gremh.



9. Subgerencia de Prestaciones Energéticas, Minera e Hidrocarburos:

Se encarga de asegurar la prestación de servicios energéticos, mineros e hidrocarburos de calidad en una organización o entidad

10. Subgerencia de Facilitación de la Calidad Energética Minera e

Hidrocarburos: Se encarga de promover y facilitar la mejora continua de la calidad en los procesos y servicios relacionados con la energía, la minería y los hidrocarburos en una organización o entidad.

El área de archivos de la Gremh almacena una variedad de documentos que se organizan siguiendo la estructura jerárquica de la gerencia, como se muestra en la **Figura 1**. Esta práctica permite una organización más eficiente de la información, dividiéndola en tres niveles, de acuerdo al organigrama de la Gremh.

Figura 2

Logo de la Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco)



Nota. Logo Gremh, por Sitio web Facebook – Cusco (<https://goo.su/Z2eKloN>)

Ubicación

La ubicación de la Gremh se encuentra en Av. Confraternidad 408, en el distrito de Wánchaq de la ciudad del Cusco, Perú. En las imágenes mostradas, se observa tanto la fachada **Figura 3** como el interior de la Gremh, **Figuras 4 y 5**.



Figura 3

Vista Frontal de la Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco)



Figura 4

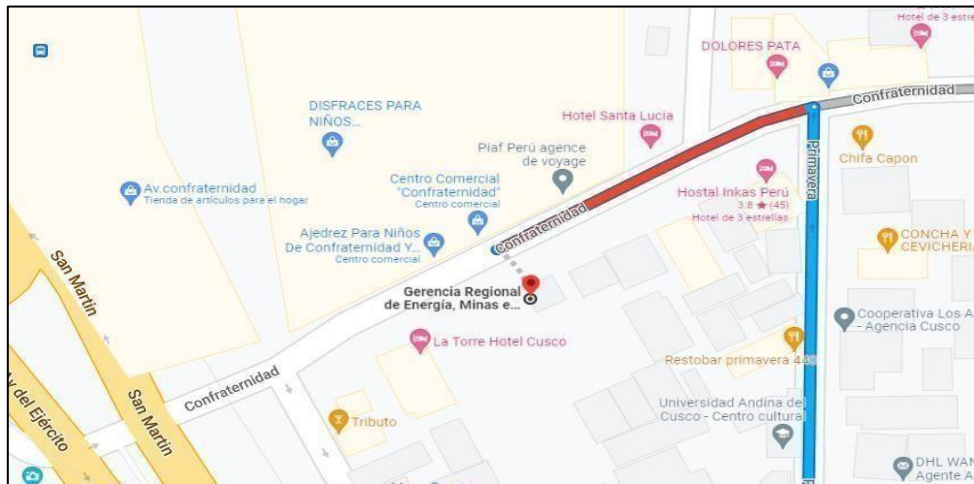
Vista Interna de la Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco)





Figura 5

Mapa de la localización de la Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco)



Nota. Mapa Gremh, por Aplicación de Google Maps (<https://goo.su/gbln>)

1.2.2 Descripción del problema

El problema que enfrenta la Gremh en la oficina de Administración, específicamente en el módulo de archivos, es la ineficiencia en la gestión de archivos y documentación. El módulo de archivos almacena una gran cantidad de documentos importantes relacionados con las industrias energética, minera e hidrocarburos. Esto resalta la necesidad crítica de contar con un sistema de gestor de archivos efectivo que facilite el almacenamiento, organización y acceso rápido y eficiente a esta información. Actualmente, se utiliza un sistema manual que ha quedado obsoleto con el paso del tiempo. El espacio disponible ya no es suficiente para almacenar toda la información, lo cual ha llevado a que los documentos estén desordenados y sea difícil encontrar y acceder a la información necesaria a continuación, se presenta los puntos específicos de la problemática:

- **Cantidad de documentos almacenados:**

La cantidad de documentos almacenados no se puede determinar con precisión debido a que la información acumulada abarca varios años y gestiones anteriores, lo que ha provocado un desorden en la ubicación de los documentos.

- **Clasificación de los documentos:**

Los documentos se clasifican de acuerdo a las áreas de la Gremh a las que están asociados. Cada área emplea ficheros de colores para señalar la



procedencia de cada documento, lo que simplifica su localización, como se aprecia en las Figuras 6, 7 y 8

Figura 6

Ficheros Color Azul de Gerencia General



Figura 7

Ficheros Color Verde de Informes de Fiscalización





Figura 8

Ficheros Color Blanco de área de Hidrocarburos



- **Método de búsqueda:**

Según la entrevista realizada, no hay un responsable específico en el módulo de archivos. En cambio, el área de administración es la encargada de autorizar el acceso al módulo de archivos a cualquier personal que lo solicite.

- **Tiempo promedio para buscar información:**

No se puede determinar un tiempo promedio para buscar información, ya que solo el personal autorizado por el área de administración puede acceder al módulo de archivos, y cada persona es responsable de buscar la información que necesita.

- **Solicitantes de la información:**

Principalmente, los solicitantes de información son las diferentes áreas de la Gremh. Estas áreas revisan documentos como informes emitidos, oficios emitidos, memorándums y expedientes técnicos. También se solicita información por parte de la Contraloría, con el fin de evaluar el trabajo realizado por la Gremh durante el período de gestión.



- **Espacio físico utilizado:**

Actualmente, el espacio físico disponible se ha sobrepasado, lo que ha llevado a que algunos documentos deban ser almacenados en el piso debido a la falta de espacio en los estantes, se puede ver el lugar de módulo de archivos en la **Figura 9**

- **Fuente de la entrevista:**

Se realizó una entrevista para obtener más detalles sobre el módulo de archivos de la Gremh. La entrevista puede encontrarse en el **Anexo 1** en las **Figuras 75, 76 ,77** y en el siguiente enlace:

(<https://goo.su/jnvz>)

El problema abordado es la insuficiencia y obsolescencia del sistema actual de gestión de archivos de manera manual en el área de archivos de la Gremh. Este problema se manifiesta en varios aspectos, como la falta de un sistema organizado para la clasificación y acceso a documentos, la acumulación desordenada de información a lo largo de los años, la limitada capacidad física de almacenamiento y la ausencia de un método eficiente para buscar información.

Figura 9

Módulo de archivos actual de la Gremh (Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco)





1.2.3 Formulación del problema

¿Cómo optimizar la administración de documentos en el área de archivos de la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco a través de la implementación de un sistema de gestor de archivos basado en la tecnología Bootstrap para la gestión eficiente de archivos?

1.2.4 Objetivos

Objetivo General

Implementar un sistema de gestor de archivos basado en la tecnología Bootstrap en el área de archivos de la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco

Objetivos Específicos

- Identificar requisitos del usuario a través de entrevistas con usuarios y equipo de la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco.
- Diseñar y desarrollar el sistema de gestor de archivos usando la tecnología Bootstrap, siguiendo un enfoque iterativo e incremental para adaptarse a las necesidades del usuario.
- Realizar el despliegue del sistema de gestor de archivos usando la tecnología Bootstrap en la Gerencia Regional de Energía, Minas e Hidrocarburos – Cusco.



1.2.5 Justificación

La necesidad de la Gremh en el área de archivos de contar con una herramienta tecnológica que permita una gestión más eficiente, segura y organizada de los documentos relacionados con las actividades de las áreas mencionadas, la implementación de un sistema de gestor de archivos permitirá optimizar la gestión documental, reduciendo tiempos y costos asociados a la gestión física de los documentos, garantizando, la integridad y la seguridad de la información y mejorando la gestión documental.

1.2.6 Alcances y limitaciones

Alcance:

El sistema de gestor de archivos abarcará el área de archivos y se encargará de la organización y almacenamiento de los documentos generales de la organización, así como de los documentos específicos de cada área y subgerencia dentro de la Gremh. Incluirá documentos como informes de gestión, resoluciones, informes regionales, estadísticas, opiniones legales, contratos, planes estratégicos, presupuestos anuales, especificaciones técnicas de sistemas, inventarios, estados contables, informes de supervisión, reportes financieros, material promocional, comunicados de prensa, políticas energéticas, reglamentos, estudios de mercado, análisis de riesgos, facturas energéticas, informes de consumo, normas de calidad y certificaciones.

Limitaciones:

- El sistema de gestor de archivos no abarcará documentos de otras áreas o instituciones que no estén directamente relacionadas con la Gremh.
- La adopción del sistema por parte del personal puede ser un desafío, especialmente si no se proporciona suficiente capacitación y apoyo en el uso del sistema.
- Es importante establecer pautas claras y brindar orientación a los usuarios sobre cómo clasificar y organizar los documentos de manera uniforme, ya que se cuenta con una gran cantidad de documentos en formato físico.



CAPÍTULO 2 – Marco Teórico

2.1 Antecedentes del desarrollo, implementación o transferencia tecnológica.

2.1.1 Antecedentes Nacionales

Antecedente N°1

Según Ramos (2020) en su Tesis de grado que titula “**Aplicación web para mejorar la gestión y control documental de la municipalidad distrital de Miguel Checa - Sojo**” desarrollado en la Universidad Nacional, Piura, Tesis para optar el Título de Ingeniero Informático.

Objetivo:

El objetivo de esta investigación es analizar las problemáticas actuales en la gestión documental de la municipalidad, identificar las causas de ineficiencia y errores en el manejo de los documentos, y proponer una solución tecnológica basada en una aplicación web. El objetivo es mejorar la eficiencia y facilitar la búsqueda, consulta y registro de los documentos.

Método:

El método utilizado consistió en realizar un análisis de las problemáticas existentes en la gestión documental, identificando las principales causas de ineficiencia y errores. A partir de este análisis, se planteó una solución tecnológica basada en una aplicación web. Se emplearon diversas herramientas como Visual Studio Code, Microsoft SQL Server Management Studio 2016 y asp.net core para el backend, y se utilizó Angular-JavaScript para el Frontend. Se desarrolló una aplicación web con una interfaz intuitiva y amigable, que ofrece herramientas avanzadas de gestión y control documental.

Conclusión:

En conclusión, esta investigación se enfocó en abordar las problemáticas actuales en la gestión documental de la municipalidad, proponiendo una solución tecnológica eficiente e innovadora. La aplicación web desarrollada permite la gestión centralizada y eficiente de los documentos, mejorando la eficiencia en el manejo de los mismos. Proporciona una interfaz amigable e intuitiva para los usuarios y ofrece herramientas avanzadas de gestión y control documental, como la clasificación por categorías, el seguimiento en tiempo real y la generación automática de informes. Esta solución tecnológica contribuye a optimizar el proceso de gestión documental, mejorando la eficiencia y reduciendo errores, lo que a su vez beneficia a la municipalidad en términos de organización y toma de decisiones.



Antecedente N°2

Según Gutiérrez (2019) en su Tesis de grado que titula “**Automatización de la gestión archivística para optimizar los pedidos de documentos del área de archivo de la Ugel 03**” desarrollado en la Universidad Científica del Sur, Lima, Trabajo de investigación para optar el Grado Académico de Bachiller en Ingeniería Empresarial y de Sistemas.

Objetivo:

El objetivo de esta investigación es diseñar y desarrollar un sistema automatizado para la gestión y control de documentos en el área de archivo de la Ugel 03. Se busca mejorar la eficiencia en el manejo de documentos y en los procesos de solicitud y entrega, facilitando la búsqueda, consulta y registro de los mismos

Método:

El método utilizado consistió en analizar las problemáticas actuales en la gestión documental del área de archivo de la Ugel 03, identificando las causas de ineficiencia en el manejo de documentos y en los procesos de solicitud y entrega. A partir de este análisis, se propuso una solución tecnológica basada en un sistema automatizado. Se utilizaron herramientas como Java como lenguaje de programación, MySQL como gestor de bases de datos, NetBeans como entorno de desarrollo integrado (IDE) y un servidor local para el desarrollo del sistema.

Conclusión:

En conclusión, esta investigación se enfocó en abordar las problemáticas existentes en la gestión documental del área de archivo de la Ugel 03, proponiendo una solución tecnológica innovadora y eficiente. El sistema automatizado desarrollado permite la gestión centralizada y eficiente de los documentos, mejorando la eficiencia en su manejo y agilizando los procesos de solicitud y entrega. Ofrece una interfaz amigable e intuitiva para los usuarios y herramientas avanzadas de gestión documental, como la clasificación por categorías y la identificación en tiempo real. La implementación de este sistema automatizado contribuye a mejorar la gestión documental en el área de archivo de la Ugel 03, optimizando los procesos y agilizando las tareas relacionadas con los documentos.

Antecedente N°3

Según Aquino (2019) en su Tesis de grado que titula “**Desarrollo e Implementación del Sistema de Gestión de Documentos ScanViewer Para la Empresa Global Factoring S.A.**” desarrollado en la Universidad Tecnológica del Perú, Lima, Para obtener el Título Profesional de: Ingeniero de Sistemas e Informática



Objetivo:

El objetivo de esta investigación es diseñar, desarrollar e implementar un sistema de gestión de documentos para la empresa Global Factoring S.A. Se busca mejorar la eficiencia en el manejo de documentos y en los procesos de almacenamiento y búsqueda, facilitando la gestión centralizada y rápida de los mismos.

Método:

El método utilizado consistió en analizar las problemáticas actuales en la gestión documental de la empresa, identificando las causas de ineficiencia en el manejo de documentos y en los procesos de almacenamiento y búsqueda. A partir de este análisis, se propuso una solución tecnológica basada en el sistema de gestión de documentos ScanViewer. Se emplearon herramientas como C# (.Net Framework), Microsoft Visual Studio, SQLServer, MS Vicio y MS Word para el desarrollo del sistema.

Conclusión:

En conclusión, esta investigación se enfocó en abordar las problemáticas existentes en la gestión documental de Global Factoring S.A, proponiendo una solución tecnológica innovadora y eficiente. El sistema de gestión de documentos ScanViewer permite la digitalización y gestión centralizada de los documentos de la empresa, mejorando la eficiencia en su manejo y agilizando los procesos de almacenamiento y búsqueda. Ofrece herramientas avanzadas de gestión documental, como la clasificación por categorías y la identificación en tiempo real, además de integrarse con otros sistemas de la empresa. La implementación de este sistema contribuye a optimizar los procesos de gestión documental, mejorar la eficiencia y rapidez en la búsqueda y consulta de documentos, así como reducir errores y garantizar la seguridad de la información de la empresa.

2.1.2 Antecedentes Internacionales

Antecedente N°1

Según Yugcha (2021) en su Tesis de maestría que titula **“Sistema de gestión documental binaria para la información gerencial de la Escuela de Conducción San Miguel Drive del cantón Salcedo.”** desarrollado en la Universidad Técnica de Cotopaxi, Ecuador, Para obtener el para obtener el grado de magíster en Maestría en Administración de Empresas.

Objetivo:

El objetivo de esta tesis es implementar un sistema de gestión documental binario (físico y electrónico) en la Escuela de Conducción "San Miguel Drive" del cantón Salcedo. El propósito es optimizar la información gerencial y mejorar la eficiencia, eficacia y



productividad de las actividades documentales de la institución.

Método:

Se llevó a cabo una investigación descriptiva y exploratoria para caracterizar e identificar los distintos documentos generados diariamente en la escuela. Además, se propone aplicar las normas de calidad internacional ISO 30300, 30301 y 15489 para mejorar los procesos documentales.

Conclusión:

En conclusión, la implementación de un sistema de gestión documental binario en la Escuela de Conducción "San Miguel Drive" es una solución práctica y efectiva para optimizar la información gerencial. La investigación realizada permitió comprender los diferentes documentos generados y adaptar el diseño del sistema de gestión documental a las necesidades específicas de la institución. Se espera que el sistema de gestión documental binario mejore la productividad de la escuela y facilite el manejo digital de la información. Esto conducirá a una mejora continua, mayor efectividad, rapidez, productividad y rentabilidad para la Escuela de Conducción.

Antecedente N°2

Según Catalina (2020) en su Tesis de maestría que titula **“Implementación de un sistema de gestión documental electrónico en la UNLP.”** desarrollado en la Universidad Nacional de la Plata, Argentina, Para obtener el para obtener el grado de magíster en especialista en Gestión de la Educación Superior.

Objetivo:

El objetivo de esta tesis es implementar un Sistema de Gestión Documental Electrónico en la Universidad Nacional de La Plata (UNLP) con el propósito de transformar los expedientes de papel en expedientes electrónicos. Se busca incorporar las Tecnologías de la Información y la Comunicación a los trámites administrativos para modernizar la administración de la institución y mejorar su capacidad de gestión.

Método:

Para lograr este objetivo, se realizó un análisis de conceptos clave y se estudiaron experiencias en instituciones similares. Con base en este análisis, se propuso una intervención que tiene como finalidad mejorar la capacidad de administración y gestión de la Universidad. Se busca agilizar trámites, ahorrar recursos y promover criterios ecológicos en la gestión de documentos.



Conclusión:

En conclusión, la implementación de un Sistema de Gestión Documental Electrónico en la Universidad Nacional de La Plata es una iniciativa importante para modernizar y mejorar la capacidad de gestión de la institución. La transformación de los expedientes de papel en expedientes electrónicos permitirá una mayor eficiencia en los trámites administrativos, así como un ahorro de recursos y una promoción de criterios ecológicos. El análisis de experiencias en instituciones similares garantiza que la propuesta se adapte adecuadamente a la realidad de la UNLP. En general, esta tesis ofrece una propuesta relevante y actual para mejorar la gestión de documentos en una institución compleja como la UNLP.

Antecedente N°3

Según Cardona y Osorio (2019) en su Tesis de grado que titula “**Diseño e Implementación de un Sistema de Gestión Documental para el cumplimiento de la ley 1712 de 2014**” desarrollado en la Universidad Tecnológica de Pereira, Colombia, Para obtener el Título Profesional de: Ingeniero de Sistemas y Computación.

Objetivo:

El objetivo de esta investigación es diseñar e implementar un sistema de gestión documental que cumpla con los requisitos de la Ley de Transparencia y Acceso a la Información Pública (Ley 1712 de 2014) en Colombia. El sistema permitirá a las organizaciones organizar, conservar y acceder a la información de manera eficiente y transparente, cumpliendo con la normativa establecida

Método:

Para lograr este objetivo, se desarrolló el sistema de gestión documental utilizando principalmente C# y ASP.Net Core. Se emplearon tecnologías web como Html, CSS, JavaScript, JQuery y Semantic UI para la interfaz y funcionalidades del sistema. Debido a restricciones presupuestarias, se utilizó Visual Studio 2019 Community Edition y otros editores gratuitos para el desarrollo del sistema.

Conclusión:

En conclusión, esta investigación se centra en la creación de un sistema de gestión documental que permita a las organizaciones cumplir con los requisitos de la Ley de Transparencia y Acceso a la Información Pública en Colombia. El objetivo final es diseñar e implementar un sistema que cumpla con los requisitos legales y se ajuste a las necesidades de las organizaciones interesadas en cumplir con la normativa. La utilización de tecnologías como C#, ASP.Net Core y diversas herramientas web



garantiza un sistema eficiente y transparente para la gestión de documentos. Esta investigación contribuye a promover la transparencia y el acceso a la información en las organizaciones, cumpliendo con las exigencias legales y facilitando la gestión documental de manera efectiva.

2.2 Bases teórico – científicos

2.2.1 Teorías sobre gestión y control documentario

Teoría de las tres edades o ciclo vital del documento

Es la designación que se le otorga a las distintas etapas o fases por las que atraviesa un documento, desde su creación hasta su eliminación o selección para su custodia permanente. Según Quispe y Vílchez (2017) se cita a Heredia (1991) quien planteó en 1972 la teoría de las tres etapas de los documentos, que resultó en diversas categorías de archivos. Esta teoría está relacionada con la noción de que los documentos no son entidades estáticas o inactivas, sino que tienen una vida propia. El objetivo de esta teoría era asegurar la presencia del profesional y los métodos aplicados para que los documentos reciban el tratamiento adecuado en cada una de las tres etapas. Este ciclo, por el cual todos los documentos atraviesan, se divide en tres etapas:

- **Etapa administrativa o activa:** Se refiere al período activo de un documento. Comienza desde su creación y pasa por diferentes etapas en las cuales es consultado, transferido y compartido con frecuencia.
- **Etapa intermedia o semi activa:** Es el período en el que el documento ha perdido utilidad para la cual fue creado. Por lo tanto, no se utiliza tan activamente como en la etapa administrativa. Sin embargo, se conserva y puede ser consultado de forma ocasional.
- **Etapa inactiva o histórica:** Se refiere al período final de los archivos. Sin embargo, no todos los archivos tienen el mismo destino. Dependiendo de su naturaleza, pueden convertirse en históricos o ser destruidos.

Los archivos históricos son aquellos que tienen valor cultural o de investigación.

Durante esta etapa, se aplican métodos de preservación para conservarlos de la manera más íntegra posible.

Estas son algunas de las etapas principales que atraviesa un documento a lo largo de su vida útil:



- **Creación:** Consiste en la generación del documento, ya sea en formato impreso o digital.
- **Almacenamiento:** Dependiendo del formato en el que se haya creado el documento, se almacena físicamente o en formato digital. En algunos casos, también se considera la etapa de digitalización cuando forma parte del proceso.
- **Categorización:** Se refiere a la organización, clasificación o indexación de los documentos según los parámetros establecidos en cada contexto.
- **Transferencia:** Consiste en el envío o entrega de un documento según sus características.
- **Distribución:** Esta etapa se refiere a documentos que tienen un uso o necesidad pública. Después de ser creados, necesitan ser difundidos a un grupo específico de personas.
- **Uso colaborativo:** En la actualidad, los archivos pueden ser utilizados y editados de manera compartida. Esta etapa es reciente y se aplica exclusivamente a documentos digitales.
- **Archivado o destrucción:** Esta es la etapa final en el ciclo de vida de todos los documentos. En este momento, se decide si el documento tiene suficiente valor histórico como para ser conservado o si debe ser destruido.

Teoría de la Macro Evaluación

Según Fenoglio (2013) la evaluación a gran escala de documentos es una teoría y práctica en el campo archivístico que permite seleccionar el material de archivo, teniendo en cuenta las funciones y relaciones internas y externas del ente creador. Solo al final del proceso se realiza la revisión de la documentación propiamente dicha. La utilización de esta evaluación a gran escala en los documentos municipales es recomendable y conveniente, ya que, aunque esta responsabilidad recae en archiveros de diferentes tipos de organismos, tanto públicos como privados. El objetivo de esta teoría es cambiar el enfoque tradicional de evaluación basado en el contenido de los documentos, por uno más amplio o a escala macro, que muestre las funciones de los entes creadores, sus programas, acciones y negociaciones, es decir, resaltar la conexión entre el documento y su motivo de creación. Para evaluar según este enfoque, se realiza un análisis del contexto de producción de los documentos, examinando tres aspectos:



- Primero, las funciones, subfunciones, programas y actividades de la entidad.
- Segundo, la estructura y el organigrama administrativo en el que se desarrollan las funciones.
- Tercero, el resultado de la interacción entre la ejecución de las funciones o programas de la institución, desarrollados a partir de una estructura administrativa.

Características de la evaluación a gran escala:

- Evaluación de todos los documentos creados en su conjunto.
- Consideración del valor social de los documentos.
- Transición de la evaluación de documentos individuales a la evaluación basada en un análisis funcional de los sistemas administrativos.
- Interrelación entre gobierno, sociedad e individuos.
- Investigación a gran escala descendente.
- Participación del archivero en el proceso desde las primeras etapas de la creación de documentos.

Teoría de Gestión de archivos

- **Historia:**

Según Díaz (2009) la teoría de la gestión de archivos nace como respuesta a la necesidad de organizar, preservar y acceder a la información documental de manera eficiente. A medida que las sociedades fueron generando un creciente volumen de registros y documentos, surgió la necesidad de desarrollar métodos y principios para administrar esta información de manera adecuada.

Los orígenes de la teoría de gestión de archivos se remontan a la antigüedad, cuando las civilizaciones más antiguas comenzaron a utilizar sistemas rudimentarios de archivo para mantener registros y documentos importantes. Sin embargo, fue a lo largo del tiempo, especialmente en la Edad Media y el Renacimiento, que se empezaron a establecer principios y técnicas más sistemáticas. En el siglo XIX, con el advenimiento de la administración moderna y la burocracia, se desarrollaron métodos más organizados de clasificación y conservación de documentos. A medida que las instituciones y organizaciones crecían, se hizo evidente la necesidad de establecer normas y prácticas estandarizadas para la gestión de archivos. En el siglo XX, se dieron importantes avances en la gestión de archivos con la aparición de nuevas tecnologías y sistemas de almacenamiento. La microfilmación, por ejemplo,



permitió la preservación de documentos en formatos reducidos y facilitó su acceso. Con la llegada de la era digital, los sistemas de gestión de documentos electrónicos (EDMS) y sistemas de gestión de contenido empresarial (ECM) se convirtieron en herramientas esenciales para la administración y preservación de documentos electrónicos.

- **Definición:**

Según Ruiz (2001) la gestión de archivos se refiere al conjunto de procesos y prácticas utilizados para administrar los documentos y registros en una organización. Su objetivo principal es garantizar el acceso eficiente, seguro y confiable a la información, así como asegurar su integridad y preservación a lo largo del tiempo. Algunos aspectos importantes de la gestión de archivos incluyen la creación, captura, organización, almacenamiento, conservación, acceso y disposición de los documentos.

La gestión de archivos implica la implementación de políticas, procedimientos y herramientas que faciliten la clasificación, indexación y búsqueda de los documentos, asegurando su disponibilidad para los usuarios autorizados. También se ocupa de la seguridad y confidencialidad de la información, estableciendo controles para proteger los archivos de accesos no autorizados y pérdidas accidentales o deliberadas:

Según Ruiz (2001) la gestión de archivos tiene los siguientes elementos:

- **Principios de archivística:** Los principios de archivística son fundamentos teóricos que guían la gestión de archivos. Incluyen conceptos como el principio de procedencia (los documentos deben organizarse según su origen), el principio de orden original (mantener el orden original de los documentos), y el principio de integridad (preservar la unidad y la autenticidad de los documentos).
- **Ciclo de vida de los documentos:** El ciclo de vida de los documentos describe las diferentes etapas por las que atraviesan los documentos desde su creación hasta su disposición final. Estas etapas incluyen la creación, la clasificación, la conservación, el acceso y la disposición de los documentos. Comprender el ciclo de vida es esencial para establecer políticas y procedimientos de gestión de archivos efectivos.



- **Organización y clasificación de documentos:** La organización y clasificación de documentos implica establecer un sistema estructurado para categorizar y etiquetar los documentos de manera lógica y coherente. Esto facilita su búsqueda, recuperación y conservación a largo plazo. Los métodos de organización pueden incluir sistemas de clasificación alfabética, numérica, temática o cronológica, entre otros.
- **Preservación y conservación de documentos:** La preservación y conservación de documentos se refiere a las medidas y técnicas utilizadas para garantizar la integridad y la longevidad de los documentos a lo largo del tiempo. Esto implica la protección contra factores físicos, químicos y biológicos que podrían dañar los documentos, así como la implementación de prácticas de almacenamiento adecuadas y el uso de materiales y técnicas de conservación apropiadas.
- **Acceso y difusión de documentos:** El acceso y la difusión de documentos se refieren a la disponibilidad y la divulgación de los documentos archivados. Esto implica establecer políticas y procedimientos para el acceso controlado a los documentos, facilitar la consulta por parte de los usuarios autorizados y promover la difusión de la información contenida en los documentos a través de medios digitales, publicaciones u otros medios de difusión.

Seguridad de información en especial de archivos digitalizados

Según Voutssas (2010) se basa en una serie de aspectos fundamentales que se enfocan en proteger la confidencialidad, integridad, disponibilidad y autenticidad de los archivos digitales.

A continuación, se presenta una definición de cada uno de estos aspectos:

- **Confidencialidad:** La confidencialidad se refiere a garantizar que la información contenida en los archivos digitalizados solo sea accesible por personas autorizadas. Para lograrlo, se utilizan técnicas de cifrado y controles de acceso para proteger los archivos de accesos no autorizados y evitar la divulgación de información sensible.
- **Integridad:** La integridad implica asegurar que los archivos digitalizados no sean alterados de manera no autorizada y que la información se mantenga exacta y



completa. Se utilizan técnicas como firmas digitales y funciones hash para verificar la integridad de los archivos y detectar cualquier modificación no autorizada.

- **Disponibilidad:** La disponibilidad se relaciona con garantizar que los archivos digitalizados estén accesibles cuando sea necesario. Esto implica implementar medidas de protección contra fallos, como copias de seguridad, redundancia de datos y sistemas de recuperación de desastres, para asegurar que los archivos estén disponibles en todo momento.
- **Autenticidad:** La autenticidad se refiere a la verificación de la identidad y origen de los archivos digitalizados. Implica utilizar técnicas de autenticación, como firmas digitales y certificados digitales, para asegurar que los archivos provengan de fuentes confiables y no hayan sido modificados o falsificados.

2.2.2 Teorías sobre desarrollo web

Metodología Programación extrema (XP)

Según Centeno y Cordones (2016) mencionan a Somerville (2011) para señalar que la programación extrema (XP) es el método ágil más reconocido y ampliamente utilizado para el desarrollo de software. El término fue acuñado debido a que este enfoque lleva las prácticas reconocidas, como el desarrollo iterativo, a niveles "extremos".

También Hernández (2014) afirma que XP es una metodología ágil que pone énfasis en la mejora de las relaciones interpersonales como clave para el éxito en el desarrollo de software. Promueve el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores y proporciona un entorno laboral favorable.

Por otro lado, Rozo (2014) sostiene que la metodología XP es una disciplina relativamente joven para el desarrollo de software. Ha demostrado su éxito en numerosas empresas de diferentes tamaños e industrias en todo el mundo, ya que pone énfasis en la satisfacción del cliente. Esta metodología está diseñada para ofrecer el software adecuado a las solicitudes de los clientes de manera oportuna. XP capacita a sus desarrolladores para responder con confianza a las cambiantes necesidades de los clientes, incluso al final del ciclo de vida.



Programación Extrema (XP) presenta las siguientes características:

- **Comunicación constante:** Fomenta la comunicación directa y constante entre los miembros del equipo de desarrollo, así como con los clientes o usuarios finales del software.
- **Desarrollo incremental:** El desarrollo se divide en pequeñas iteraciones o incrementos, donde se entregan funcionalidades completas en cada ciclo. Esto permite obtener retroalimentación temprana y ajustar el desarrollo según las necesidades del cliente.
- **Pruebas unitarias:** Se enfatiza la escritura de pruebas unitarias automatizadas para garantizar la calidad del código y facilitar su mantenimiento. Estas pruebas se ejecutan frecuentemente durante el desarrollo.
- **Integración continua:** Se promueve la integración regular y frecuente del código desarrollado por diferentes miembros del equipo. Esto permite detectar y corregir problemas de integración de forma temprana.
- **Diseño simple:** Se busca mantener el diseño del software lo más simple posible, evitando la incorporación de funcionalidades innecesarias. Esto facilita la comprensión y el mantenimiento del código.
- **Refactorización:** Se fomenta la mejora continua del código a través de la refactorización. Esto implica reestructurar y optimizar el código sin cambiar su comportamiento externo, con el objetivo de mejorar su calidad y mantenibilidad.
- **Orientación al cliente:** El cliente o usuario final es una parte fundamental en el proceso de desarrollo. Se busca una interacción cercana y constante con el cliente para comprender y satisfacer sus necesidades y expectativas.
- **Equipo autónomo y colaborativo:** Se fomenta la colaboración y la autogestión dentro del equipo de desarrollo. Los miembros trabajan de manera conjunta y toman decisiones en conjunto para lograr los objetivos establecidos.
- **Entrega continua:** El software se entrega de forma continua y frecuente, lo que permite obtener retroalimentación temprana y realizar ajustes en función de las necesidades y prioridades del cliente.



- **Adaptabilidad y flexibilidad:** La metodología XP se adapta fácilmente a los cambios, ya que está diseñada para acomodar nuevos requerimientos o modificaciones en cualquier etapa del desarrollo.

Tecnología Bootstrap

Según Jakobus y Marah (2017), Es una herramienta de diseño web que permite la creación de interfaces responsivas y adaptables a diferentes dispositivos. La tecnología Bootstrap se ha convertido en una de las más utilizadas en el desarrollo de aplicaciones web debido a su facilidad de uso y su gran cantidad de componentes predefinidos.

La motivación detrás de Bootstrap era proporcionar una herramienta que simplificara el proceso de diseño y desarrollo de interfaces web. En ese momento, el diseño responsive y la compatibilidad con múltiples dispositivos se estaban volviendo cada vez más importantes. Además, había una necesidad de estandarizar y agilizar el desarrollo de interfaces visuales coherentes en Twitter.

La tecnología Bootstrap fue creada por dos desarrolladores de Twitter, Mark Otto y Jacob Thornton. Nació como una solución interna para ayudar en el desarrollo de proyectos web en Twitter. En agosto de 2011, lanzaron la primera versión de Bootstrap de forma pública y de código abierto.

Según Jakobus y Marah (2017), Bootstrap se basó en varias tecnologías web populares de la época, como HTML, CSS y JavaScript. Proporcionando una biblioteca de componentes reutilizables, estilos predefinidos y utilidades de diseño que los desarrolladores podían utilizar para crear rápidamente interfaces atractivas y responsivas. La comunidad de desarrolladores web comenzó a adoptar Bootstrap como una herramienta estándar para el desarrollo front-end debido a su facilidad de uso, flexibilidad y diseño responsivo.

A lo largo de los años, Bootstrap ha evolucionado con actualizaciones y nuevas versiones que agregan características y mejoras. La comunidad de desarrolladores ha contribuido a su crecimiento y ha creado una amplia gama de complementos y extensiones que amplían aún más sus capacidades.

La tecnología Bootstrap ha revolucionado el desarrollo web al proporcionar a los desarrolladores una base sólida y consistente para crear interfaces modernas y responsivas de manera eficiente.

Según Jakobus y Marah (2017) Bootstrap tiene los siguientes elementos:

- **Framework front-end:** Bootstrap es un framework front-end de código abierto que proporciona un conjunto de herramientas y estilos predefinidos para el



desarrollo de sitios web y aplicaciones. Está basado en HTML, CSS y JavaScript, y ofrece una serie de componentes y utilidades que facilitan la creación de interfaces responsivas y modernas.

- **Diseño responsive:** Bootstrap se enfoca en el desarrollo de sitios web y aplicaciones con diseño responsive, lo que significa que se adaptan automáticamente a diferentes tamaños de pantalla y dispositivos, como computadoras de escritorio, tablets y teléfonos móviles. Esto se logra mediante el uso de una rejilla flexible y componentes que se ajustan de manera fluida a los distintos tamaños de pantalla.
- **Componentes y estilos predefinidos:** Bootstrap incluye una amplia variedad de componentes y estilos predefinidos, como botones, barras de navegación, formularios, carruseles, tarjetas, entre otros. Estos componentes están diseñados de manera consistente y estilizada, lo que permite ahorrar tiempo y esfuerzo en la creación de interfaces visuales atractivas y coherentes.
- **Personalización y extensibilidad:** A pesar de contar con componentes y estilos predefinidos, Bootstrap ofrece flexibilidad para personalizar y extender su apariencia y funcionalidad. Mediante el uso de variables CSS y clases modificadoras, es posible ajustar el aspecto visual de los componentes según las necesidades del proyecto. Además, se pueden agregar complementos y extensiones para ampliar las capacidades de Bootstrap.
- **Compatibilidad entre navegadores:** Bootstrap se desarrolla con la compatibilidad entre navegadores en mente. Esto significa que está diseñado para funcionar correctamente en una amplia gama de navegadores web, incluyendo Chrome, Firefox, Safari, Edge, e Internet Explorer. Al utilizar Bootstrap, puedes tener la confianza de que tu sitio web o aplicación se verá y funcionará de manera consistente en diferentes plataformas y navegadores.

Arquitectura cliente-servidor

Esta arquitectura distribuye la carga de trabajo entre un cliente y uno o más servidores. El cliente realiza solicitudes al servidor para acceder y manipular los archivos, mientras que el servidor se encarga del almacenamiento y la gestión de los mismos. Esta arquitectura es escalable y permite un procesamiento eficiente de las solicitudes las capas son las siguientes:



- **Capa de presentación** (interfaz de usuario): Esta capa se encarga de mostrar la interfaz gráfica al usuario y permite la interacción con el sistema. Utilizando tecnología Bootstrap, se pueden diseñar y desarrollar interfaces modernas y responsivas, con componentes y estilos predefinidos que facilitan la creación de una experiencia de usuario agradable.
- **Capa de aplicación:** En esta capa se encuentra la lógica de la aplicación, incluyendo el procesamiento de eventos y la manipulación de datos. Aquí se definen las funcionalidades del gestor de archivos, como la creación, eliminación, búsqueda y organización de archivos. También se pueden implementar reglas de negocio específicas del sistema.
- **Capa de datos:** Esta capa se encarga de la gestión y almacenamiento de los datos del sistema de archivos. Incluyendo la conexión a una base de datos o sistemas de almacenamiento de archivos, donde se guardan y recuperan los datos de los archivos, así como otra información relevante, como metadatos y permisos de acceso.



CAPÍTULO 3 – Desarrollo, Implementación o Transferencia Tecnológica

3.1. Detalles del Proyecto

La aplicación web responsiva de design contendrá los siguientes módulos: un módulo de inicio de sesión para los usuarios, un módulo de registro de usuarios para manejar múltiples usuarios, un menú horizontal y un conjunto de archivos y herramientas para el desarrollo. El apartado de "herramientas para el desarrollo" se refiere a las herramientas que se utilizarán durante el proceso de desarrollo en sí. Todos estos aspectos están relacionados con las reglas de negocio del proyecto. Por ejemplo, la regla de negocio de tener un inicio de sesión para evitar el acceso directo, o la elección de un menú horizontal en lugar de uno vertical. Es importante revisar cada punto detalladamente para aclarar cualquier duda.

- **Login de usuario:** El login de usuario se utiliza para iniciar sesión en la aplicación mediante un nombre de usuario único y una contraseña. Esto permite gestionar los datos específicos de cada usuario. Es importante que la aplicación no permita la repetición de nombres de usuario, es decir, no debe permitir que se ingresen usuarios con el mismo nombre. Esto garantiza la integridad y la seguridad de la información de cada usuario.
- **Registro de usuario para múltiples usuarios:** Este módulo se utiliza para registrar a los múltiples usuarios que utilizarán el sistema. Es importante para garantizar un control adecuado de acceso. Durante el registro, se solicitará información básica del usuario, como nombre, apellidos, DNI, correo electrónico, nombre de usuario y contraseña. Es crucial que la contraseña esté encriptada para mantener la seguridad de los datos almacenados en la base de datos. Al obtener esta información, se podrá gestionar la base de datos y determinar el funcionamiento del sistema.
- **Menú horizontal:** El menú es un módulo fundamental que permite la navegación entre los diferentes módulos de la aplicación web. Debe incluir opciones como "Inicio", "Agregar archivos" y "Salir del sistema". Gracias al menú, los usuarios podrán moverse de manera intuitiva y acceder rápidamente a las funcionalidades que necesiten dentro de la aplicación. Es una herramienta clave para facilitar la experiencia de uso y la interacción con los diferentes componentes del sistema.



- **CRUD de archivos:** El CRUD de archivos es el módulo más importante de un sistema que permite realizar las operaciones básicas de Create (Crear), Read (Leer), Update (Actualizar) y Delete (Eliminar).
 - a) En primer lugar, al crear un nuevo registro, se agrega un archivo que está vinculado a la base de datos. El archivo se guarda en una carpeta del servidor y se envía un registro con la información correspondiente a la base de datos.
 - b) La parte de lectura implica visualizar la información de los archivos en una tabla, donde se muestran todos los registros.
 - c) En la actualización, se pueden realizar cambios en el nombre del archivo o en alguna fecha, aunque normalmente se prefiere eliminar y sobrescribir el archivo en lugar de actualizarlo directamente.

En cuanto a las herramientas recomendadas para el desarrollo, se llevarán a cabo en el sistema operativo Windows, y como navegador web se utilizará Chrome, con su respectiva consola de desarrollo.

Se menciona el uso de Apache Server como servidor web. Las siglas "XAMPP" representan lo siguiente: X se refiere a cualquier sistema operativo, ya sea Windows, Linux, macOS, etc. A es Apache, que es un servidor web ampliamente utilizado. M es MySQL, que es un sistema de gestión de bases de datos relacional. P es PHP, que es un lenguaje de programación popularmente utilizado para el desarrollo web. Y P es Perl, que es otro lenguaje de programación utilizado principalmente en el ámbito web.

3.1.2 Metodología XP

A continuación, en la **Figura 10 y Tabla 1** se presentan los pasos principales de la metodología XP que se aplicarán en el proyecto de desarrollo del sistema de gestor de archivos para la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco en el área de archivos.



Figura 10

Fases de la metodología XP (Programación Extrema)

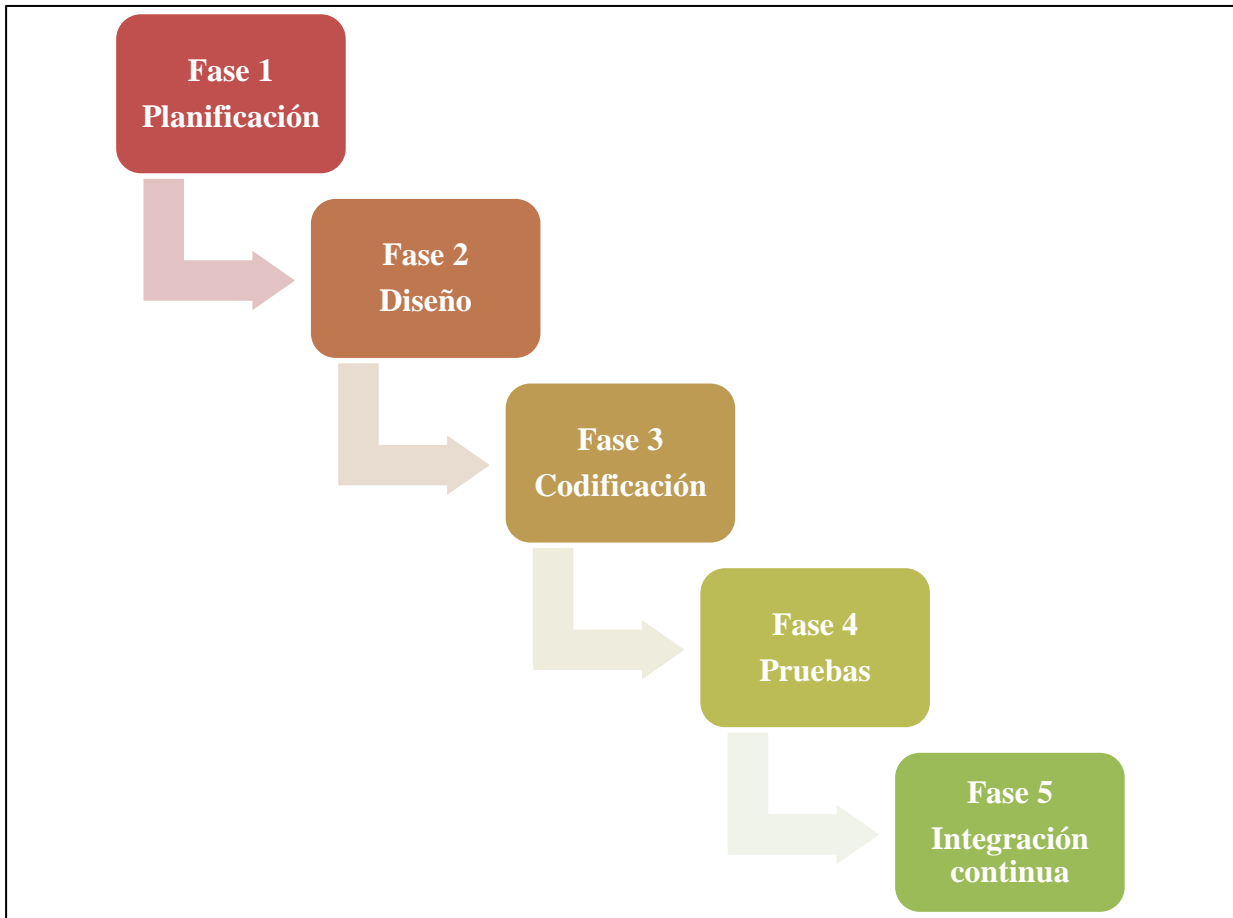




Tabla 1

Los pasos principales de la metodología XP en el proyecto.

SISTEMA DE GESTOR DE ARCHIVOS BASADO EN LA TECNOLOGÍA BOOTSTRAP PARA LA GERENCIA REGIONAL DE ENERGÍA, MINAS E HIDROCARBUROS - CUSCO EN EL ÁREA DE DOCUMENTOS.

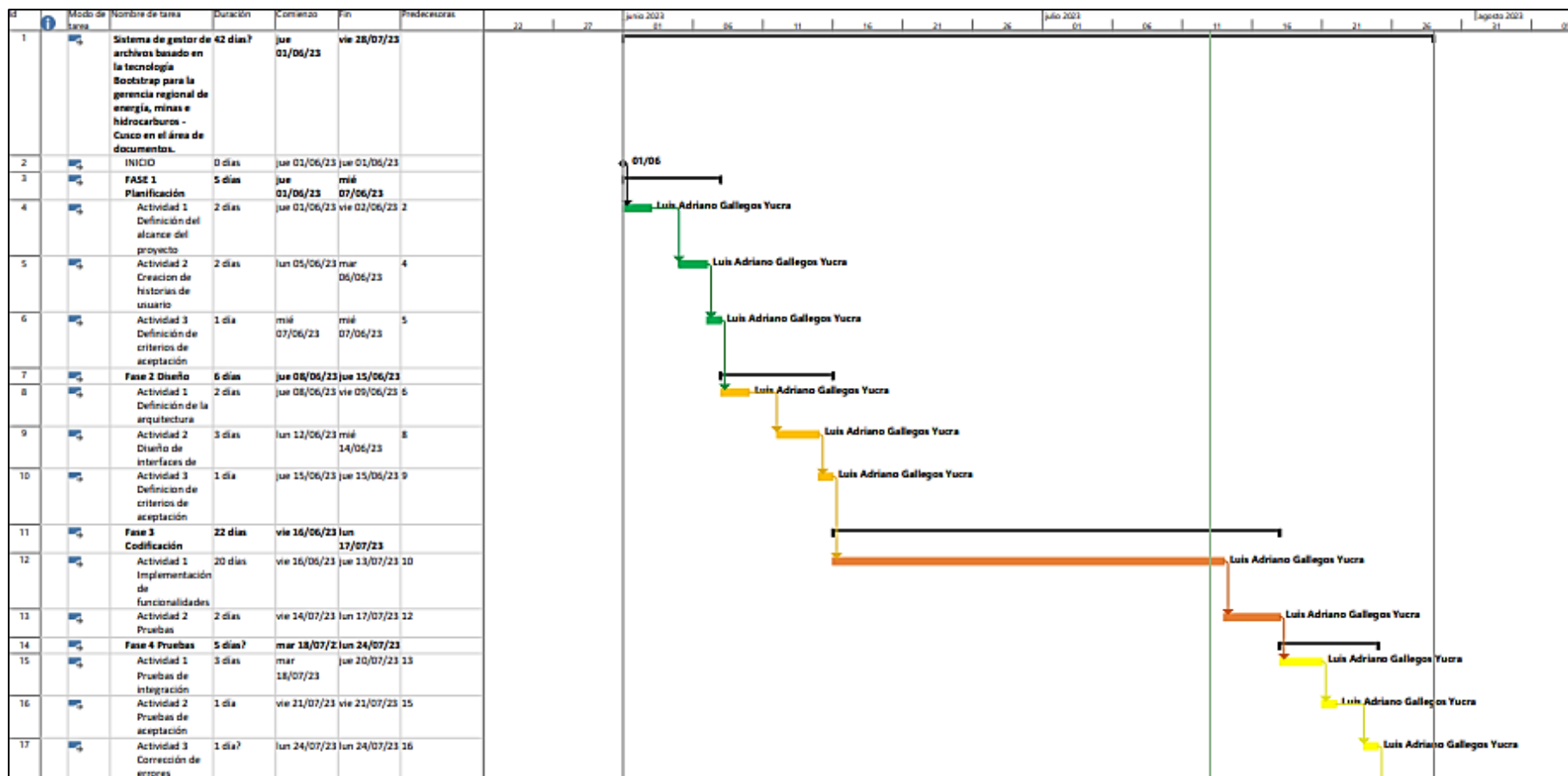
Metodología XP				
Fase 1	Fase 2	Fase 3	Fase 4	Fase 5
Planificación:	Diseño:	Codificación:	Pruebas:	Integración continua:
En esta fase, se define el alcance del proyecto y se establecen las historias de usuario que describen las funcionalidades del sistema de gestor de archivos. También se definen los criterios de aceptación para cada historia de usuario y se crea un plan de lanzamiento para el sistema	En esta fase, se define la arquitectura del sistema de gestor de archivos y se establecen las interfaces de usuario. También se establecen los criterios de aceptación para cada historia de usuario y se diseñan las pruebas unitarias y de aceptación.	En la fase de codificación, se implementan las funcionalidades del sistema de gestor de archivos utilizando las mejores prácticas de programación. Como desarrollador trabajar en horas para mejorar la calidad del código y se realizan pruebas unitarias para cada funcionalidad implementada.	En esta fase, se realizan pruebas de integración para verificar que todas las funcionalidades del sistema de gestor de archivos funcionen correctamente juntas. También se realizan pruebas de aceptación para cada historia de usuario y se corrigen los errores encontrados	La integración continua permite integrar el código de manera continua y detectar los errores de manera temprana. Esto reduce el tiempo necesario para corregir errores y mejorar la calidad del sistema.



Para obtener detalles sobre las actividades y requisitos para el desarrollo del sistema de gestor de archivos, se ha elaborado un presupuesto que se encuentra en el **Anexo 2**. Este presupuesto proporciona información clave sobre los recursos financieros y los costos asociados con el proyecto, lo que es fundamental para la planificación y ejecución del desarrollo del sistema., En la **Figura 11**, se muestra el Plan de trabajo en detalle de actividades.

Figura 11

Detalle de actividades Plan de trabajo de desarrollo de gestor de archivos en la Gremh (gerencia regional de energía, minas e hidrocarburos - Cusco) – Área de Documentos (<https://goo.su/qQp6PGL>)



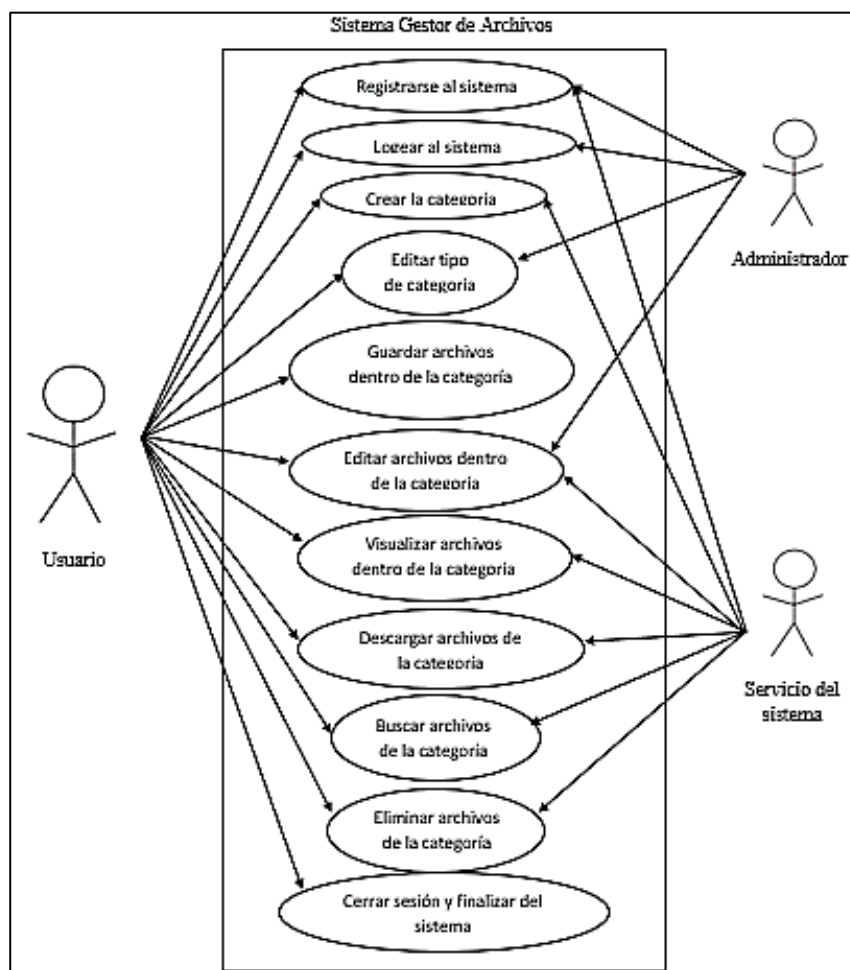


3.1.2.1 Fase 1 Planificación

1. **Definición del alcance del proyecto:** En esta actividad, se describen en detalle los requerimientos de los usuarios y las funcionalidades que debe tener el sistema de gestor de archivos. Se identificarán las funciones que cumplirán el usuario, el administrador y los servicios del sistema. A continuación, en la **Figura 12**, se presenta el diagrama de casos de uso que tendrá el sistema de gestor de archivos.

Figura 12

Diagrama de caso de uso del sistema de gestor de archivos para la Gremh



2. **Creación de historias de usuario:** En esta actividad, se han identificado cuatro casos de historias de usuario con sus respectivas necesidades y requerimientos, los cuales se detallan a continuación en las **Tablas 2, 3, 4 y 5**.



Tabla 2

Caso I de historia de usuario: Buscar archivos por nombre

Historia de usuario área Gerencia General:	Como usuario quiero poder buscar archivos por nombre para encontrarlos más fácilmente
Criterio de aceptación:	El sistema debe permitir al usuario buscar archivos por nombre, coincidiendo con el texto ingresado de manera parcial o completa. La búsqueda debe ser en mayúsculas o minúsculas, y debe mostrar resultados en tiempo real.

Tabla 3

Caso II de historia de usuario: Subir archivos al sistema

Historia de usuario Gerencia General:	Como usuario quiero poder subir archivos al sistema de gestor de archivos desde mi computadora
Criterio de aceptación:	El sistema debe permitir al usuario subir archivos al seleccionar la opción de carga en la interfaz de usuario y seleccionando el archivo que desea cargar desde su computadora. El sistema debe verificar que el archivo cumpla con los requisitos de formato y tamaño establecidos, y si el archivo no cumple con estos requisitos, se debe mostrar un mensaje de error al usuario. Después de que el archivo se carga correctamente, debe estar disponible para cualquier usuario con los permisos correspondientes. Además, el sistema debe mostrar una notificación al usuario que cargó el archivo para confirmar que la carga se realizó con éxito y registrar todas las actividades de carga de archivos para su posterior seguimiento y control.

Tabla 4

Caso III de historia de usuario: Organizar archivos en carpetas

Historia de usuario área Patrimonio:	Como usuario quiero poder organizar mis archivos en carpetas para tener una mejor estructura y orden
---	--



Criterio de aceptación:	El sistema debe permitir al usuario crear y eliminar carpetas de archivos. El usuario debe poder seleccionar la ubicación de la carpeta en el sistema. El sistema debe mostrar la estructura de carpetas de manera clara y jerárquica para una mejor organización de los archivos.
--------------------------------	--

Tabla 5

Caso IV de historia de usuario: Descargar archivos del sistema

Historia de usuario área Patrimonio:	Como usuario quiero poder descargar archivos desde el sistema de gestor de archivos para tener una copia local de los mismos
---	--

Criterio de aceptación:	El sistema debe permitir al usuario descargar un archivo seleccionando la opción de descarga en la interfaz de usuario. La descarga debe comenzar inmediatamente y mostrar el progreso de la misma. El archivo descargado debe tener el mismo contenido y formato que el archivo original en el sistema de gestor de archivos. La descarga debe estar disponible para cualquier usuario.
--------------------------------	--

3. Definición de criterios de aceptación: Se definen los casos de prueba que se utilizarán para verificar el correcto funcionamiento del sistema de gestor de archivos, tanto a nivel de componentes individuales como a nivel de la aplicación completa. Estos casos de prueba se detallan en las **Tablas 6, 7, 8 y 9**.

Tabla 6

Caso de prueba 1: Subir un archivo a una categoría específica en el sistema de gestor de archivos.

Caso de prueba:	Escenario: Subir un archivo a una categoría específica en el sistema de gestor de archivos.
------------------------	---

Pasos:	<ul style="list-style-type: none"> • Abrir la aplicación de gestor de archivos. • Navegar hasta la categoría en la que se desea subir el archivo. • Seleccionar la opción de subir un archivo.
---------------	---



-
- Seleccionar el archivo que se desea subir desde el sistema local.
 - Confirmar la acción de subir el archivo.
 - Verificar que el archivo se haya subido correctamente en la categoría especificada
-

En esta prueba de aceptación, se identifica que el sistema de gestor de archivos permite subir archivos a una categoría específica y que el archivo se carga correctamente en esa categoría.

Tabla 7

Caso de prueba 2: Eliminar un archivo de una categoría específica en el sistema de gestor de archivos.

Caso de prueba:	Escenario: Eliminar un archivo de una categoría específica en el sistema de gestor de archivos.
------------------------	---

- | | |
|---------------|---|
| Pasos: | <ul style="list-style-type: none">• Abrir la aplicación de gestor de archivos.• Navegar hasta la categoría donde se encuentra el archivo que se desea eliminar.• Identificar el archivo que se desea eliminar.• Seleccionar la opción de eliminar el archivo.• Confirmar la acción de eliminación del archivo.• Verificar que el archivo haya sido eliminado correctamente de la categoría especificada. |
|---------------|---|
-

En esta prueba de aceptación, se identifica que el sistema de gestor de archivos permite eliminar archivos de una categoría específica y que la eliminación se realiza correctamente.

Tabla 8

Caso de prueba 3: Descargar un archivo desde una categoría específica en el sistema de gestor de archivos.

Caso de prueba:	Escenario: Descargar un archivo desde una categoría específica en el sistema de gestor de archivos.
------------------------	---

Precondiciones:	Debe haber al menos un archivo presente en la categoría seleccionada.
------------------------	---



Pasos:	<ul style="list-style-type: none">• Abrir la aplicación de gestor de archivos.• Navegar hasta la categoría donde se encuentra el archivo que se desea descargar.• Identificar el archivo que se desea descargar.• Seleccionar la opción de descargar el archivo.• Verificar que se inicie la descarga del archivo correctamente.
---------------	--

En esta prueba de aceptación, se identifica que el sistema de gestor de archivos permite descargar archivos de una categoría específica.

Tabla 9

Caso de prueba 4: Visualizar los archivos en una categoría específica en el sistema de gestor de archivos.

Caso de prueba:	Escenario: Visualizar los archivos en una categoría específica en el sistema de gestor de archivos.
Precondiciones:	Deben existir archivos en la categoría seleccionada.
Pasos:	<ul style="list-style-type: none">• Abrir la aplicación de gestor de archivos.• Navegar hasta la categoría donde se encuentran los archivos que se desean visualizar.• Verificar que los archivos en la categoría sean visibles y estén correctamente listados.• Seleccionar un archivo de la lista.• Verificar que se muestre una vista previa o los detalles del archivo seleccionado.

En esta prueba de aceptación, se identifica que el sistema de gestor de archivos permite visualizar los archivos presentes en una categoría específica.

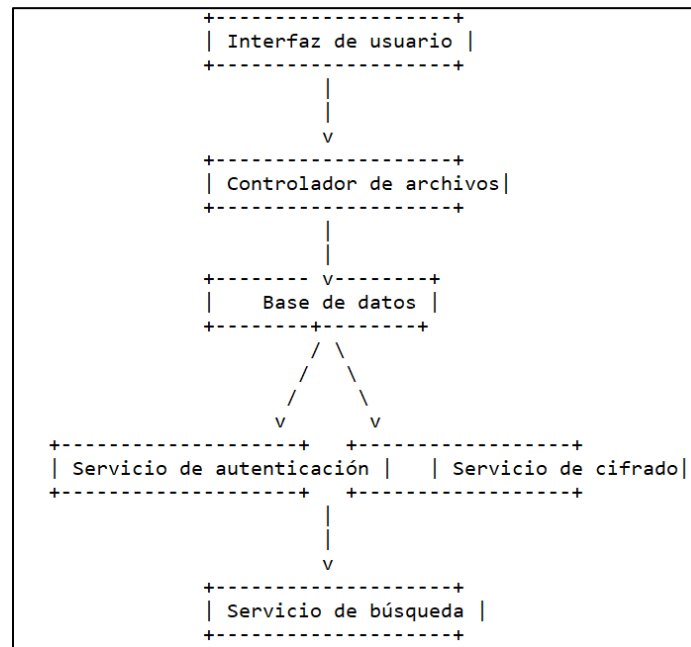
3.1.2.2 Fase 2 Diseño

1. **Diseño de la arquitectura del sistema:** En esta actividad se define la estructura y organización del sistema de gestor de archivos. Se determinan los componentes principales del sistema en la **Figura 13**. También se definen las tecnologías que se utilizarán para implementar la arquitectura. Estableciendo la estructura de cliente-

servidor en el sistema de gestor de archivos, incluyendo los componentes principales y la forma en que se relacionan entre sí.

Figura 13

Diagrama de componentes UML de la arquitectura del sistema de gestor de archivos.



También se eligen las herramientas y tecnologías que se utilizarán para desarrollar el sistema de gestor de archivos, de acuerdo a los requerimientos y objetivos del proyecto.

Como entorno tecnología principal y framework de diseño web gratuito y de código abierto se hará uso del Bootstrap, como lenguaje de programación se realizará en PHP

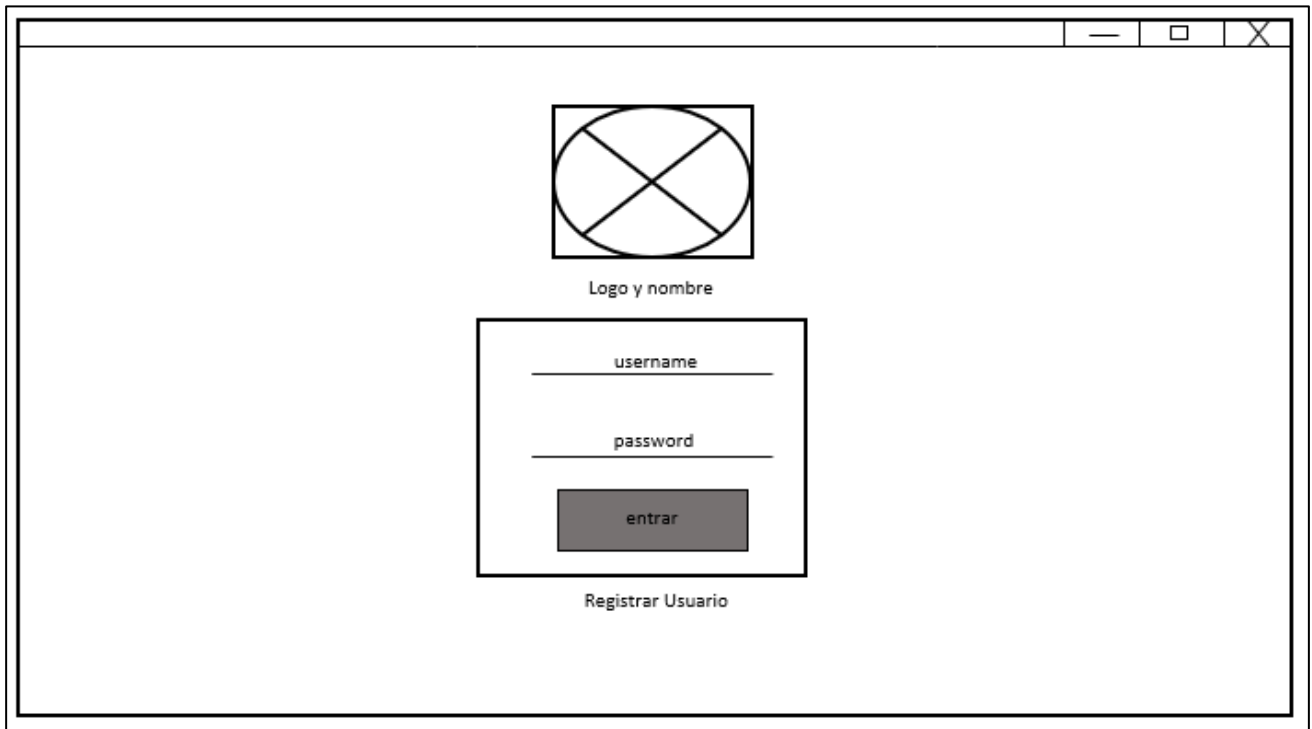
Como tecnologías utilizadas para el desarrollo web se usará JavaScript, HTML, CSS y la biblioteca jQuery en la que cada una de estas se define en las bases teóricas.

Para crear un entorno de servidor web local en un equipo de escritorio se hará uso del software libre XAMPP, para desarrollar y programar el sistema de gestor de archivos se hará uso del editor de texto Sublime Text.

- 2. Diseño de interfaces de usuario:** En esta actividad se diseñan las interfaces gráficas para la interacción de los usuarios con el sistema de gestor de archivos. Se definen las pantallas, botones y menús, que incluyen un inicio de sesión con usuario y contraseña, el logo de la gerencia, y las opciones de inicio, categorías, archivos, y salir, como se muestra en las **Figuras 14, 15, 16 y 17**.

Figura 14

Parte I maquetación (Login de usuario) de proyecto gestor de archivos Gremh.



Dentro de las Funcionalidades se encontrará 2 módulos Categoría, Archivos.

Figura 15

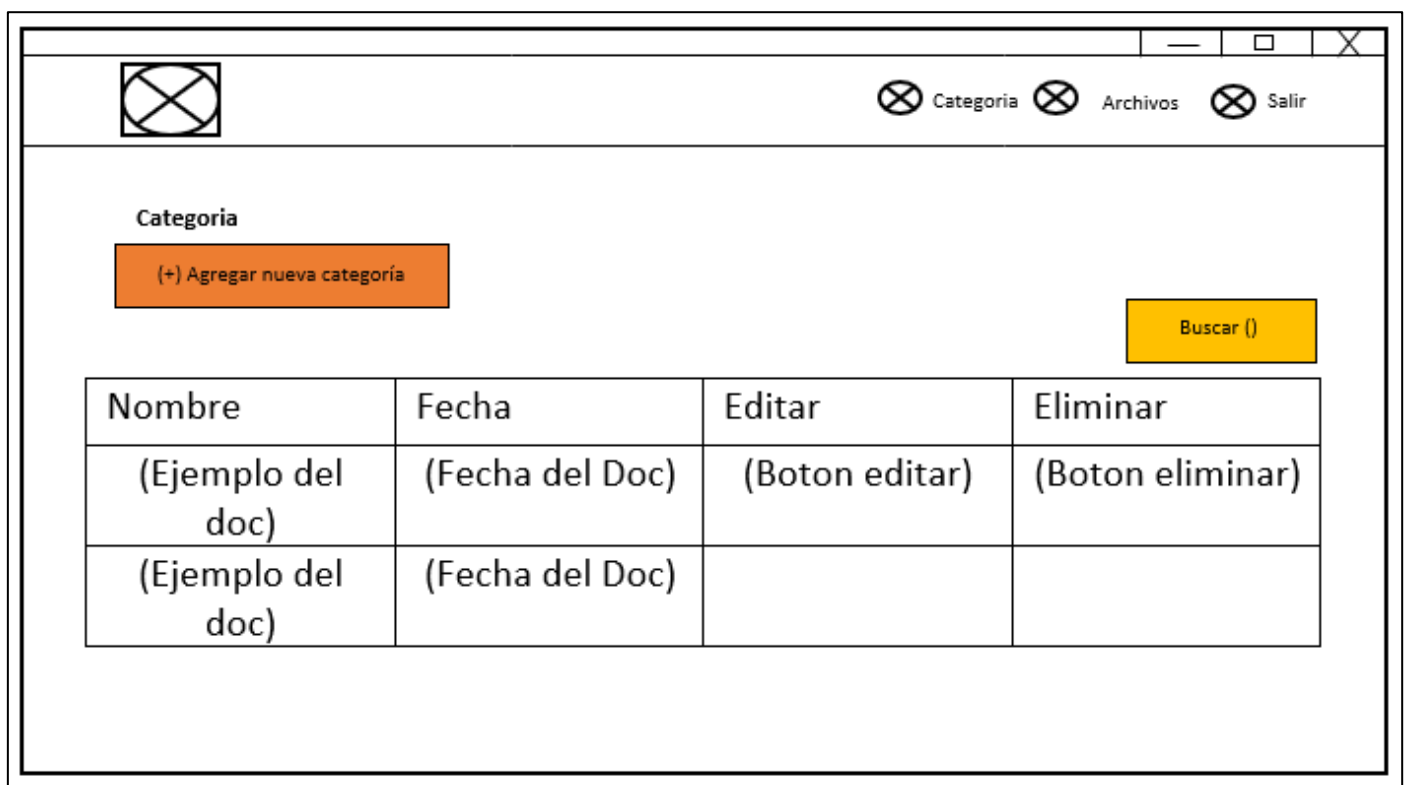
Parte II maquetación (Inicio de usuario) de proyecto gestor de archivos Gremh.



Dentro del módulo de categoría el usuario va a poder encontrar el botón agregar nueva categoría en el cual podrá colocar el nombre de la categoría que se usará como ejemplo: Informes, Memorándum o Comunicados, seguidamente se mostrará la fecha que se agregó, si por algún error está mal escrito el documento se podrá editar y finalmente hubo un error se podrá eliminar así mismo se agregará el botón buscar para hacer más fácil el acceso a la información.

Figura 16

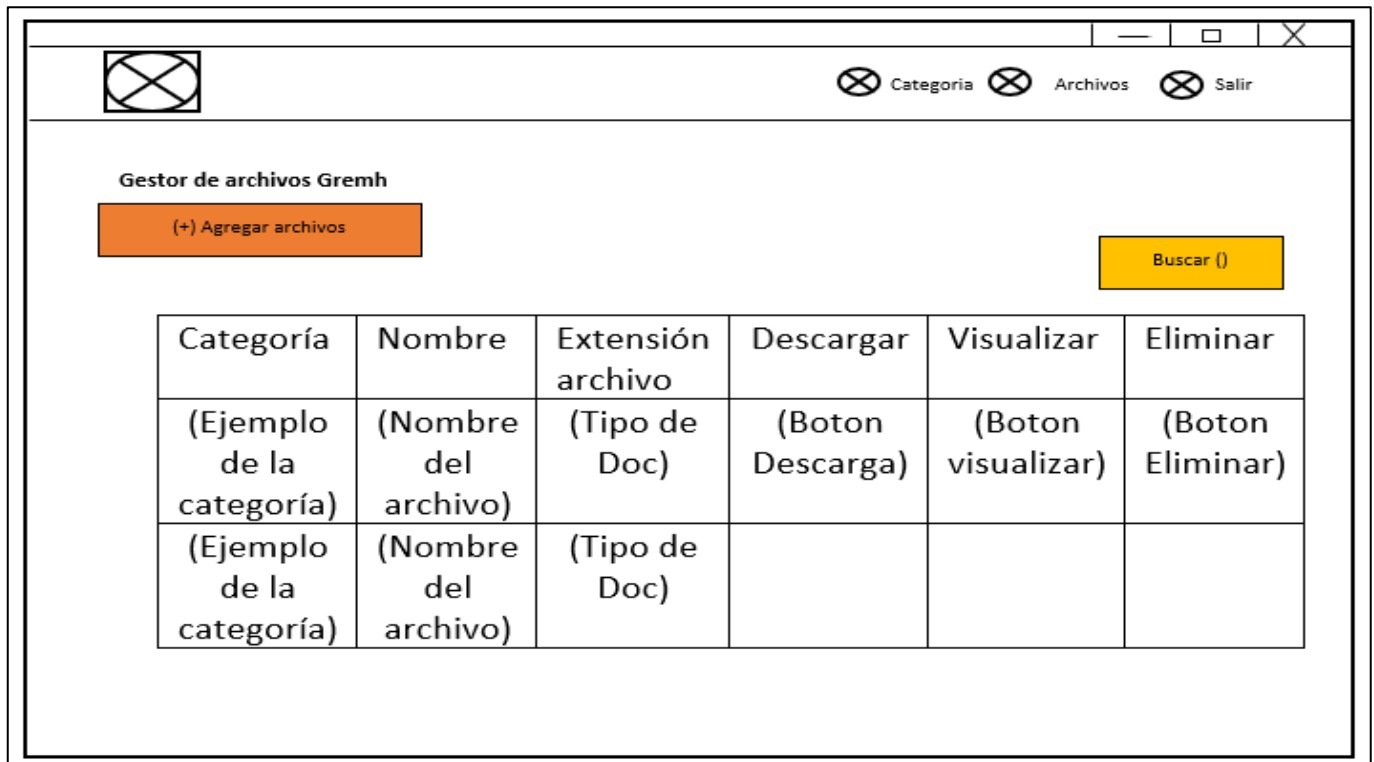
Parte III maquetación (Modulo Categoría) de proyecto gestor de archivos Gremh.



Dentro del módulo de archivo el usuario una vez agregado la categoría va a poder agregar los documentos de acuerdo a la categoría que creó, también podrá buscar el documento para tener más facilidad de encontrar el documento luego podrá visualizar el nombre del documento. En este módulo el usuario podrá descargar, visualizar y eliminar el archivo con el fin de que sea más fácil manejar los archivos. Finalmente, el usuario podrá salir del sistema en el módulo de salir esto lo redijera de nuevo al login de usuario para finalizar.

Figura 17

Parte IV maquetación (Modulo Archivo) de proyecto gestor de archivos Gremh.



3. Definición de criterios de aceptación: Se establecen los criterios que las historias de usuario deben cumplir para considerarse completas. Estos criterios se derivaron de las historias de casos de uso propuestas y las entrevistas realizadas, y se detallan en las **Tablas 2, 3, 4 y 5**. Estas tablas contienen los criterios específicos que deben cumplirse en el desarrollo del sistema de gestor de archivos, abarcando aspectos fundamentales como la historia de usuario y el criterio de aceptación.

- Una vez que los criterios estén bien definidos, se procede a la implementación de las funcionalidades necesarias, asegurando el cumplimiento de los estándares y requisitos establecidos. Esto garantizará que el sistema de gestor de archivos cumpla con las expectativas de los usuarios y se ajuste a las necesidades de la organización.
- La definición clara de estos criterios proporciona una guía sólida para avanzar en el desarrollo, permitiendo priorizar las tareas y recursos de manera efectiva. Además, contribuye a mantener un enfoque constante en la calidad y el rendimiento del sistema.

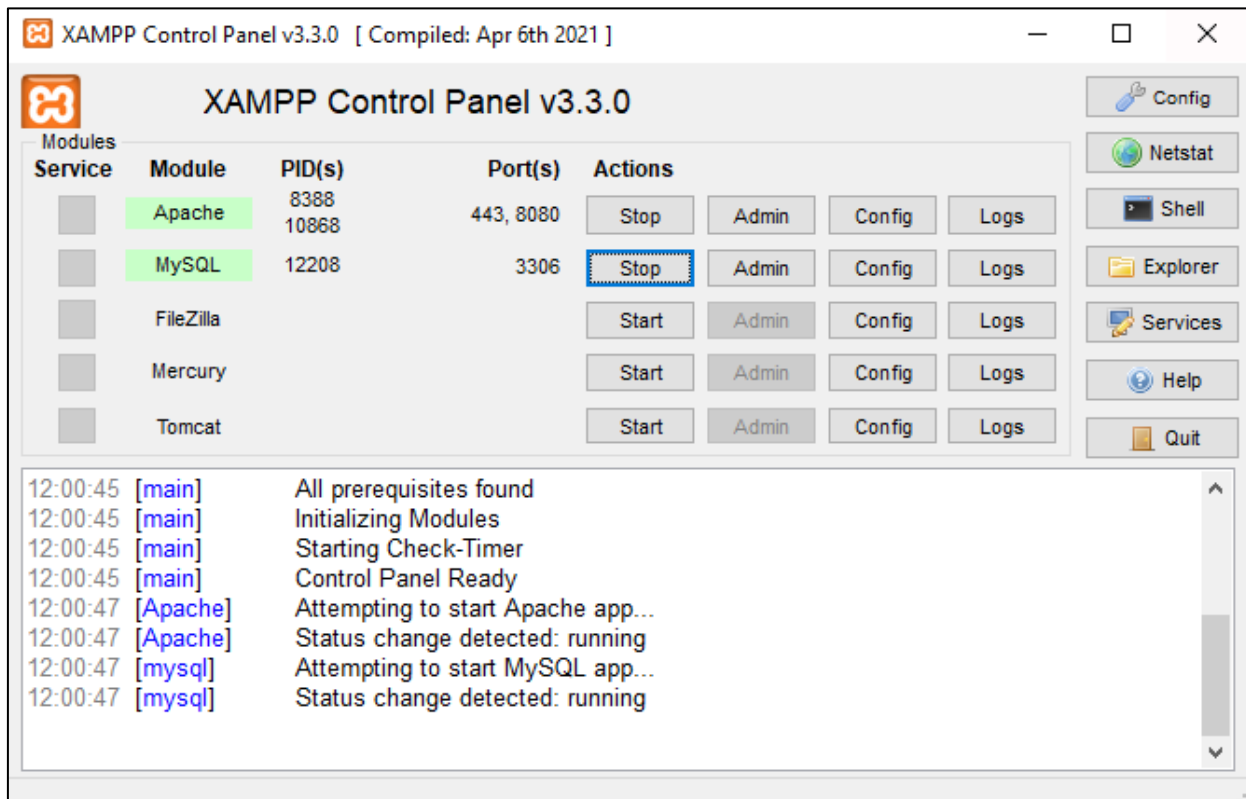


3.1.2.3 Fase 3 Codificación

1. **Implementación de las funcionalidades:** En esta actividad, se lleva a cabo la codificación de las funcionalidades definidas en las fases anteriores. Se empezará por descargar e instalar los programas ya mencionados así conjuntamente el servidor local en la **Figura 18** y la aplicación de entorno de desarrollo web, se codifican las funcionalidades del sistema de gestor de archivos.

Figura 18

Iniciar el servidor web XAMPP conjunto el Apache y al MySQL



En un esquema de trabajo previamente establecido, se identificará el progreso de cada avance, lo que permitirá una organización más eficiente en la construcción del sistema de gestor de archivos, tal como se puede apreciar en la **Figura 19** y la información detallada de cada carpeta en la **Tabla 10**.



Figura 19

Estructura de carpeta para el desarrollo del gestor de archivos

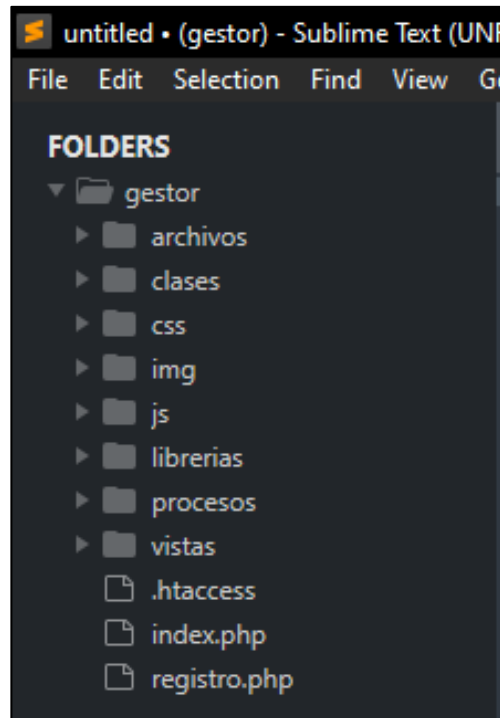


Tabla 10

Información cada carpeta de la estructura del gestor de archivos

Carpeta general "gestor" Es la carpeta principal del proyecto.	
---	--

Carpeta "clases"	Contiene las clases PHP del lado del servidor que se van a utilizar
Carpeta "css"	Contiene todos los archivos de diseño necesarios.
Carpeta "img"	Contiene los archivos de imágenes que se utilizarán para el maquetado.
Carpeta "js"	Contiene los archivos de JavaScript que contienen las funciones del proyecto
Carpeta "procesos"	Es una carpeta intermedia que simula los controladores de la programación orientada a objetos. Aquí se depuran los datos antes de enviarlos a las clases, donde los datos ya están corregidos.



Carpeta "vistas" Contiene todas las vistas dentro de la aplicación principal.

Carpeta "librerías" Contiene todas las librerías externas que se agregarán al proyecto, como por ejemplo Bootstrap.

Se inicia por la carpeta "clases", como se muestra en la **Figura 20**, que incluye los archivos: Categorías.php, Conexión.php, Gestor.php, y Usuario.php.

- **Categorías.php:** Esta clase proporciona métodos para agregar, eliminar, obtener y actualizar categorías en una base de datos utilizando consultas SQL.

Figura 20

Categorías.php, métodos para agregar, eliminar, obtener y actualizar categorías en una base de datos SQL

```
1 <?php
2
3 require_once "Conexion.php";
4 class Categorías extends Conectar {
5     public function agregarCategoria($datos) {
6         $conexion = Conectar::conexion();
7
8         $sql = "INSERT INTO t_categorias (id_usuario, nombre)
9             VALUES (?, ?)";
10        $query = $conexion->prepare($sql);
11        $query->bind_param("is", $datos['idusuario'],
12            $datos['categoria']);
13        $respuesta = $query->execute();
14        $query->close();
15
16        return $respuesta;
17    }
18
19    public function eliminarCategorías($idCategoría) {
20        $conexion = Conectar::conexion();
21
22        $sql = "DELETE FROM t_categorias
23            WHERE id_categoria = ?";
24        $query = $conexion->prepare($sql);
25        $query->bind_param("i", $idCategoría);
26        $respuesta = $query->execute();
27        $query->close();
28        return $respuesta;
29    }
30
31    public function obtenerCategoria($idCategoría){
32        $conexion = Conectar::conexion();
33
34        $sql = "SELECT id_categoria,
35            nombre
36            FROM t_categorias
37            WHERE id_categoria = '$idCategoría'";
38        $result = mysqli_query($conexion, $sql);
39
40        $categoria = mysqli_fetch_array($result);
41
42        $datos = array(
43            "idCategoría" => $categoria['id_categoria'],
44            "nombreCategoría" => $categoria['nombre']
45        );
46        return $datos;
47    }
48
49    public function actualizarCategoria($datos){
```

- a) La función "agregarCategoria(\$datos)" se utiliza para agregar una nueva categoría a la base de datos. Toma un arreglo llamado "\$datos" como argumento, que debe contener los valores de "idUsuario" y "categoría". La función prepara una consulta SQL de inserción y ejecuta la consulta



utilizando los valores proporcionados. Devuelve la respuesta de la ejecución de la consulta.

- b) La función "eliminarCategorias(\$idCategoria)" se utiliza para eliminar una categoría de la base de datos. Toma el identificador de la categoría como argumento y prepara una consulta SQL de eliminación. Luego, ejecuta la consulta utilizando el identificador proporcionado y devuelve la respuesta de la ejecución de la consulta.
 - c) La función "obtenerCategoria(\$idCategoria)" se utiliza para obtener los detalles de una categoría específica de la base de datos. Toma el identificador de la categoría como argumento y prepara una consulta SQL de selección. Ejecuta la consulta y recupera los resultados. Luego, crea un arreglo asociativo con los datos de la categoría encontrada y lo devuelve.
 - d) La función "actualizarCategoria(\$datos)" se utiliza para actualizar el nombre de una categoría en la base de datos. Toma un arreglo llamado "\$datos" como argumento, que debe contener los valores de "categoría" y "idCategoria". La función prepara una consulta SQL de actualización y ejecuta la consulta utilizando los valores proporcionados. Devuelve la respuesta de la ejecución de la consulta.
- **Conexión.php:** En la **Figura 21**, se presenta el código que define una clase llamada "Conectar", la cual incluye un método diseñado para establecer una conexión con una base de datos utilizando MySQL.

Figura 21

Establecer una conexión con una base de datos utilizando la extensión MySQL.

```
C:\xampp\htdocs\gestor\clases\Conexion.php (gestor) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  gestor
    archivos
    clases
      Categorías.php
      Conexión.php
      Gestor.php
      Usuario.php
    css
    img
Conexion.php
1 <?php
2
3
4 class Conectar{
5     public function conexion(){
6         $Conexion= new mysqli("localhost","root","","gestorgremh");
7         if($Conexion->connect_error){
8             echo "Falló al conectar :". $Conexion->connect_error;
9         }
10
11         $Conexion->set_charset("utf8mb4");
12         return $Conexion;
13     }
```



- a) La clase "Conectar" contiene un único método llamado "conexión ()". Este método se utiliza para establecer una conexión con la base de datos. Dentro del método, se crea una instancia de la clase "mysqli" pasando los parámetros del host (localhost), el usuario de la base de datos (root), la contraseña (vacía en este caso) y el nombre de la base de datos (gestorgremh).
 - b) Después de crear la instancia de "mysqli", el código verifica si hubo algún error al conectar con la base de datos utilizando la propiedad "connect_error" de la instancia. Si se produce un error, se muestra un mensaje de error.
 - c) A continuación, se establece el conjunto de caracteres de la conexión a "utf8mb4" utilizando el método "set_charset()". Esto asegura que la conexión esté configurada para admitir caracteres multibyte.
 - d) Finalmente, el método devuelve la instancia de "mysqli" que representa la conexión establecida.
- **Gestor.php:** En la **Figura 22**, se presenta la definición de una clase denominada "Gestor" que hereda de la clase "Conectar". Esta clase se emplea para llevar a cabo operaciones relacionadas con el gestor de archivos en la base de datos. La clase "Gestor" proporciona métodos para agregar, obtener, eliminar y mostrar archivos en la base de datos utilizando consultas SQL, además de generar el código HTML correspondiente a cada tipo de archivo.

Figura 22

Operaciones relacionadas con el manejo de archivos en la base de datos

```
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

class Gestor extends Conectar {
    public function agregarRegistroArchivo($datos){
        $conexion = Conectar::conexion();
        $sql = "INSERT INTO t_archivos (id_usuario,
            id_categoria,
            nombre,
            tipo,
            ruta)
            VALUES (?, ?, ?, ?, ?)";
        $query = $conexion->prepare($sql);
        $query->bind_param("iiss", $datos['idUsuario'],
            $datos['idCategoria'],
            $datos['nombreArchivo'],
            $datos['tipo'],
            $datos['ruta']);
        $respuesta = $query->execute();
        $query->close();
        return $respuesta;
    }

    public function obtenerNombreArchivo($idArchivo) {
        $conexion = Conectar::conexion();
        $sql = "SELECT nombre
            FROM t_archivos
            WHERE id_archivo = '$idArchivo'";
        $result = mysqli_query($conexion, $sql);
        return mysqli_fetch_array($result)['nombre'];
    }

    public function eliminarRegistroArchivo($idArchivo){
        $conexion = Conectar::conexion();
        $sql = "DELETE FROM t_archivos WHERE id_archivo = ?";
        $query = $conexion->prepare($sql);
        $query->bind_param('i', $idArchivo);
        $respuesta = $query->execute();
        $query->close();
        return $respuesta;
    }
}
```




```
47 public function obtenerRutaArchivo($idArchivo) {
48     $conexion = Conectar::conexion();
49
50     $sql = "SELECT nombre, tipo
51           FROM t_archivos
52           WHERE id_archivo = '$idArchivo'";
53     $result = mysqli_query($conexion, $sql);
54     $datos = mysqli_fetch_array($result);
55     $nombreArchivo = $datos['nombre'];
56     $extension = $datos['tipo'];
57     return self::tipoArchivo($nombreArchivo, $extension);
58 }
59
60
61 public function tipoArchivo($nombre, $extension){
62     $idUsuario = $_SESSION['idusuario'];
63
64     $ruta = "../archivos/".$idUsuario."/".$nombre;
65     switch ($extension) {
66         case 'png':
67             return '';
68             break;
69
70         case 'jpg':
71             return '';
72             break;
73         case 'pdf':
74             return '<embed src="'. $ruta . '"#toolbar=0&navpanes=0&scrollbar=0" type="application/pdf" width="100%" height="600px" />';
75             break;
76
77         case 'mp4':
78             return '<video src="'. $ruta. '" controls width="100%" height="600px"></video>';
79             break;
80
81         case 'mp3':
82             return '<audio controls src="'. $ruta. '"></audio>';
83             break;
84
85         default:
86             // ...

```

- a) La función "agregaRegistroArchivo(\$datos)" se utiliza para agregar un nuevo registro de archivo a la base de datos. Toma un arreglo llamado "\$datos" como argumento, que debe contener los valores de "idUsuario", "idCategoria", "nombreArchivo", "tipo" y "ruta". La función prepara una consulta SQL de inserción y ejecuta la consulta utilizando los valores proporcionados. Devuelve la respuesta de la ejecución de la consulta.
- b) La función "obtenNombreAchivo(\$idArchivo)" se utiliza para obtener el nombre de un archivo específico de la base de datos. Toma el identificador del archivo como argumento y prepara una consulta SQL de selección. Ejecuta la consulta y recupera el nombre del archivo encontrado. Luego, devuelve el nombre del archivo.
- c) La función "eliminaRegistroArchivo(\$idArchivo)" se utiliza para eliminar un registro de archivo de la base de datos. Toma el identificador del archivo como argumento y prepara una consulta SQL de eliminación. Luego, ejecuta la consulta utilizando el identificador proporcionado y devuelve la respuesta de la ejecución de la consulta.
- d) La función "obtenerRutaArchivo(\$idArchivo)" se utiliza para obtener la ruta de un archivo específico de la base de datos. Toma el identificador del archivo como argumento y prepara una consulta SQL de selección. Ejecuta la consulta y recupera el nombre del archivo y su extensión. Luego, llama al método "tipoArchivo()" para obtener el tipo de archivo y devuelve el resultado.



- e) La función "tipoArchivo(\$nombre, \$extension)" se utiliza para determinar el tipo de archivo y generar el código HTML correspondiente para mostrarlo en la página. Toma el nombre del archivo y su extensión como argumentos. Dependiendo de la extensión del archivo, se genera y devuelve el código HTML adecuado para mostrar imágenes, archivos PDF, archivos de video y archivos de audio.
- **Usuario.php:** La clase mencionada se utiliza para llevar a cabo operaciones relacionadas con los usuarios en la base de datos. En la **Figura 23**, se pueden encontrar métodos que permiten agregar usuarios, buscar usuarios duplicados y llevar a cabo el proceso de inicio de sesión de usuarios en la base de datos utilizando consultas SQL.

Figura 23

Métodos para agregar usuarios, buscar usuarios repetidos y realizar el inicio de sesión de usuarios en la base de datos.

```
1 <?php
2
3 require_once "Conexion.php";
4
5 class Usuario extends Conectar{
6
7     public function agregarUsuario($datos){
8         $conexion = Conectar::conexion();
9         if (self::buscarUsuarioRepetido($datos['usuario'])) {
10             return 2;
11         }else {
12             $sql = "INSERT INTO t_usuarios (nombre,
13                                     apellidos,
14                                     dni,
15                                     email,
16                                     usuario,
17                                     password)
18                                     VALUES (?, ?, ?, ?, ?, ?)";
19             $query = $conexion->prepare($sql);
20             $query->bind_param('ssssss', $datos['nombre'],
21                               $datos['apellidos'],
22                               $datos['dni'],
23                               $datos['email'],
24                               $datos['usuario'],
25                               $datos['password']);
26             $exito = $query->execute();
27             $query->close();
28             return $exito;
29         }
30     }
31
32 }
33
34 public function buscarUsuarioRepetido($usuario){
35     $conexion = Conectar::conexion();
36     $sql = "SELECT usuario
37           FROM t_usuarios
38           WHERE usuario = '$usuario'";
39     $result = mysqli_query($conexion, $sql);
40     $datos = mysqli_fetch_array($result);
41
42     if ($datos['usuario'] != "" || $datos['usuario'] == $usuario) {
43         return 1;
44     }else {
45         return 0;
46     }
47 }
```



```
48 public function login($usuario, $password) {
49     $conexion = Conectar::conexion();
50
51     $sql = "SELECT count(*) as existeUsuario
52           FROM t_usuarios
53           WHERE usuario = '$usuario'
54             AND password = '$password'";
55     $result = mysqli_query($conexion, $sql);
56
57     $respuesta = mysqli_fetch_array($result)['existeUsuario'];
58
59     if ($respuesta > 0) {
60         $_SESSION['usuario'] = $usuario;
61
62         $sql = "SELECT idt_usuario
63               FROM t_usuarios
64               WHERE usuario = '$usuario'
65                 AND password = '$password'";
66         $result = mysqli_query($conexion, $sql);
67         $idUserio = mysqli_fetch_row($result)[0];
68         $_SESSION['idUserio'] = $idUserio;
69
70         return 1;
71     } else {
72         return 0;
73     }
74 }
75 }
76
77 ?>
```

- a) La función "agregarUsuario(\$datos)" se utiliza para agregar un nuevo usuario a la base de datos. Toma un arreglo llamado "\$datos" como argumento, que debe contener los valores de "nombre", "apellidos", "dni", "email", "usuario" y "password". La función verifica si el usuario ya existe llamando al método "buscarUsuarioRepetido()". Si el usuario ya existe, devuelve 2. Si el usuario no existe, prepara una consulta SQL de inserción y ejecuta la consulta utilizando los valores proporcionados. Devuelve el resultado de la ejecución de la consulta.
- b) La función "buscarUsuarioRepetido(\$usuario)" se utiliza para verificar si un usuario ya está registrado en la base de datos. Toma el nombre de usuario como argumento y prepara una consulta SQL de selección. Ejecuta la consulta y verifica si se encontraron datos coincidentes. Si se encuentra un usuario coincidente, devuelve 1, de lo contrario, devuelve 0.
- c) La función "login(\$usuario, \$password)" se utiliza para realizar el inicio de sesión de un usuario. Toma el nombre de usuario y la contraseña como argumentos. Prepara una consulta SQL de selección para verificar si el usuario y la contraseña coinciden en la base de datos. Si se encuentra una coincidencia, establece una variable de sesión "usuario" con el nombre de usuario y otra variable de sesión "idUserio" con el identificador de usuario



correspondiente. Devuelve 1 si el inicio de sesión es exitoso, de lo contrario, devuelve 0.

Dentro de la segunda carpeta "Css", se encuentra el archivo "login.css", que contiene la hoja de estilos para varios elementos HTML. En la **Figura 24**, se pueden observar las características clave del código, que incluyen:

- Definición del color de fondo y la fuente utilizada en el documento HTML.
- Establecimiento de estilos para enlaces, encabezados y otros elementos.
- Definición de estilos para formularios y elementos de formulario, como botones, campos de texto y campos de contraseña.
- Creación de animaciones CSS para dar a los elementos una apariencia animada en la página.
- Establecimiento de un estilo de subrayado para los enlaces que resalta cuando se coloca el cursor sobre ellos.
- Definición de estilos adicionales para otros elementos y clases específicas.

Figura 24

Formato y diseño para la formulación y animación de la página HTML

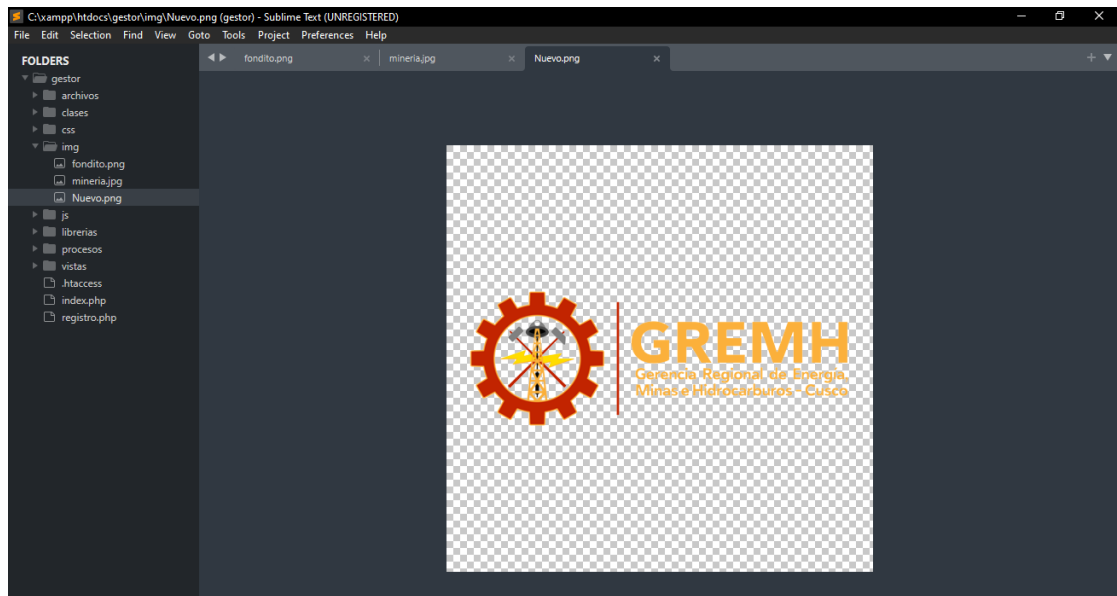
```
1 2
2  /* BASIC */
3
4  html {
5    background-color: #56baed;
6  }
7
8  body {
9    font-family: "Poppins", sans-serif;
10   height: 100vh;
11 }
12
13 a {
14   color: #92badd;
15   display: inline-block;
16   text-decoration: none;
17   font-weight: 400;
18 }
19
20 h2 {
21   text-align: center;
22   font-size: 16px;
23   font-weight: 600;
24   text-transform: uppercase;
25   display: inline-block;
26   margin: 40px 8px 10px 8px;
27   color: #cccccc;
28 }
29
30
31
32  /* STRUCTURE */
33
34 .wrapper {
35   display: flex;
36   align-items: center;
37   flex-direction: column;
38   justify-content: center;
39   width: 100%;
40   min-height: 100%;
41   padding: 20px;
42 }
43
44 #formcontent {
45   border-radius: 10px 10px 10px 10px;
46   border-radius: 10px 10px 10px 10px;
47   background: #fff;
48   padding: 30px;
49   width: 90%;
50   max-width: 450px;
51   position: relative;
```



En la tercera carpeta, denominada "Img", se encuentran los archivos en formato de imágenes, como se muestra en la **Figura 25**. Estas imágenes se utilizarán en el desarrollo del gestor de archivos.

Figura 25

Archivo en formato de imágenes.

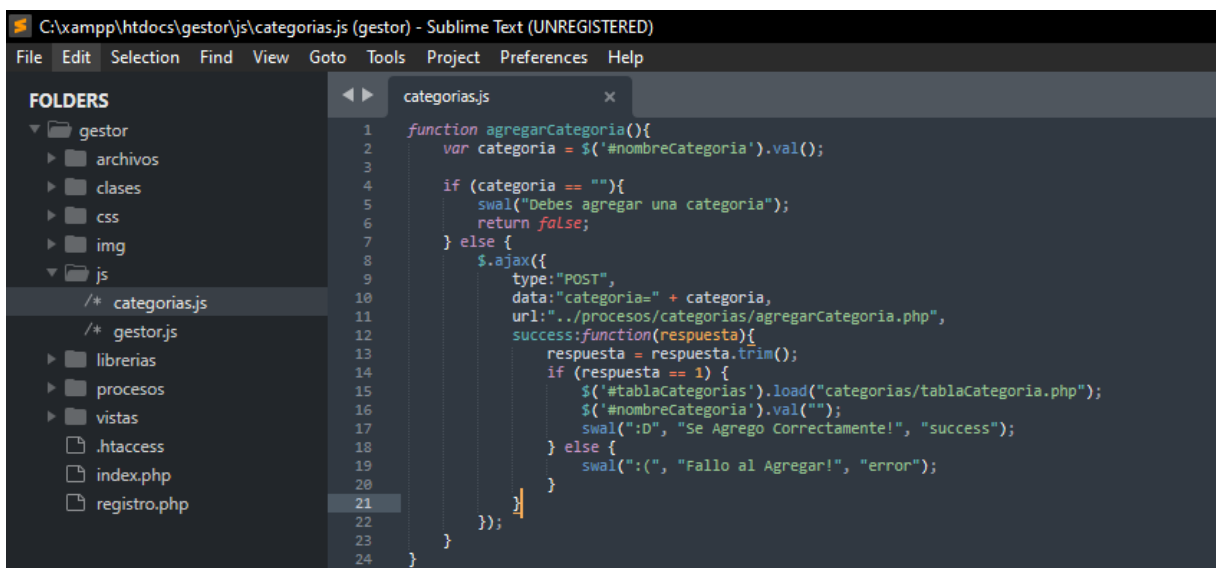


En la cuarta carpeta "js" se encuentran los archivos "categorías.js" y "gestor.js".

- **Categorías.js:** En la **Figura 26**, se encuentran funciones en JavaScript que gestionan categorías, permitiendo agregar, eliminar, obtener y actualizar categorías a través de solicitudes AJAX para interactuar con archivos PHP en el servidor.

Figura 26

Funciones para agregar, eliminar, obtener y actualizar categorías mediante el uso de solicitudes AJAX.





```
24 }
25
26 function eliminarCategorias(idCategoria){
27     idCategoria = parseInt(idCategoria);
28     if (idCategoria < 1){
29         swal("No tienes id de categoria!");
30         return false;
31     } else {
32         //*****
33         swal({
34             title: "Estas seguro de eliminar esta categoria?",
35             text: "Una vez eliminada, no podra recuperarse!",
36             icon: "warning",
37             buttons: true,
38             dangerMode: true,
39         })
40         .then((willDelete) => {
41             if (willDelete) {
42                 $.ajax({
43                     type:"POST",
44                     data:"idCategoria=" + idCategoria,
45                     url:"../procesos/categorias/eliminarCategoria.php",
46                     success:function(respuesta){
47                         respuesta = respuesta.trim();
48                         if (respuesta == 1) {
49                             $('#tablaCategorias').load("categorias/tablaCategoria.php");
50                             swal("Eliminado con Exito!", {
51                                 icon: "success",
52                             });
53                         } else {
54                             swal(":", "Fallo al eliminar!", "error");
55                         }
56                     }
57                 });
58             }
59         });
60         //*****
61     }
62 }
```

```
63 function obtenerDatosCategoria(idCategoria){
64     $.ajax({
65         type:"POST",
66         data:"idCategoria=" + idCategoria,
67         url:"../procesos/categorias/obtenerCategoria.php",
68         success:function(respuesta) {
69             respuesta = jQuery.parseJSON(respuesta);
70
71             $('#idCategoria').val(respuesta['idCategoria']);
72             $('#categoriaU').val(respuesta['nombreCategoria']);
73         }
74     })
75 }
76
77 function actualizaCategoria(){
78     if ($('#categoriaU').val() == ""){
79         swal("No hay categoria!");
80         return false;
81     } else {
82         $.ajax({
83             type:"POST",
84             data:$('#frmActualizaCategoria').serialize(),
85             url:"../procesos/categorias/actualizaCategoria.php",
86             success:function(respuesta){
87                 respuesta = respuesta.trim();
88
89                 if (respuesta == 1) {
90                     $('#tablaCategorias').load("categorias/tablaCategoria.php");
91                     swal(":", "Se actualizo con Exito!", "success");
92                 } else {
93                     swal(":", "Fallo al actualizar!", "error");
94                 }
95             }
96         });
97     }
98 }
```



- a) La función "agregarCategoria()" se utiliza para agregar una nueva categoría. Obtiene el valor de la categoría desde un campo de entrada en el formulario. Si el campo está vacío, muestra un mensaje de advertencia. De lo contrario, realiza una solicitud AJAX al servidor para agregar la categoría utilizando el archivo "agregarCategoria.php". Si la respuesta del servidor es 1, recarga la tabla de categorías, borra el campo de entrada y muestra una notificación de éxito. De lo contrario, muestra una notificación de error.
- b) La función "eliminarCategorias(idCategoria)" se utiliza para eliminar una categoría. Recibe el identificador de la categoría como argumento. Primero, muestra una ventana emergente de confirmación para verificar si el usuario desea eliminar la categoría. Si el usuario confirma la eliminación, realiza una solicitud AJAX al servidor para eliminar la categoría utilizando el archivo "eliminarCategoria.php". Si la respuesta del servidor es 1, recarga la tabla de categorías y muestra una notificación de éxito. De lo contrario, muestra una notificación de error.
- c) La función "obtenerDatosCategoria(idCategoria)" se utiliza para obtener los datos de una categoría. Recibe el identificador de la categoría como argumento. Realiza una solicitud AJAX al servidor para obtener los datos de la categoría utilizando el archivo "obtenerCategoria.php". Luego, actualiza los campos del formulario con los datos obtenidos.
- d) La función "actualizaCategoria ()" se utiliza para actualizar una categoría existente. Verifica si el campo de categoría está vacío y muestra una notificación si es así. De lo contrario, realiza una solicitud AJAX al servidor para actualizar la categoría utilizando el archivo "actualizaCategoria.php". Si la respuesta del servidor es 1, recarga la tabla de categorías y muestra una notificación de éxito. De lo contrario, muestra una notificación de error.
- **Gestor.js:** En la **Figura 27**, se encuentran funciones relacionadas con la administración de archivos en un sistema de gestión. Estas funciones posibilitan acciones como agregar, eliminar y obtener archivos en el gestor, utilizando solicitudes AJAX para comunicarse con archivos PHP en el servidor.



Figura 27

Funciones para administrar archivos mediante solicitudes AJAX.

```
gestor.js
1 function agregarArchivosGestor() {
2   var formData = new FormData(document.getElementById('frmArchivos'));
3   $.ajax({
4     url: '../procesos/gestor/guardarArchivos.php',
5     type: 'POST',
6     datatype: 'html',
7     data: formData,
8     cache: false,
9     contentType: false,
10    processData: false,
11    success: function(respuesta){
12      respuesta = respuesta.trim();
13
14      if (respuesta == 1) {
15        $('#frmArchivos')[0].reset();
16        $('#tablaGestorArchivos').load("gestor/tablaGestor.php");
17        swal(":", "Agregado con exito!", "success");
18      } else {
19        swal(":", "Fallo al agregar!", "error");
20      }
21    }
22  });
23 }
24
25 function eliminarArchivo(idArchivo){
26   swal({
27     title: "Estas seguro de eliminar este archivo?",
28     text: "Una ves eliminado, no podra recuperarse!",
29     icon: "warning",
30     buttons: true,
31     dangerMode: true,
32   })
33   .then((willDelete) => {
34     if (willDelete) {
35       $.ajax({
36         type: "POST",
37         data: "idArchivo=" + idArchivo,
38         url: "../procesos/gestor/eliminaArchivo.php",
39         success: function(respuesta){
40
41           respuesta = respuesta.trim();
42           if (respuesta == 1) {
43             $('#tablaGestorArchivos').load("gestor/tablaGestor.php");
44             swal("Eliminado con exito!", {
45               icon: "success",
46             });
47           } else {
48             swal("Error al eliminar!", {
49               icon: "error",
50             });
51           }
52         }
53       });
54     }
55   });
56 }
57
58 function obtenerArchivoPorId(idArchivo){
59   $.ajax({
60     type: "POST",
61     data: "idArchivo=" + idArchivo,
62     url: "../procesos/gestor/obtenerArchivo.php",
63     success: function(respuesta){
64       $('#archivoObtenido').html(respuesta);
65     }
66   });
67 }
68
69
70
71
```

- a) La función "agregarArchivosGestor()" se utiliza para agregar archivos al gestor. Crea un objeto FormData a partir del formulario con el ID "frmArchivos" y realiza una solicitud AJAX al servidor utilizando el archivo "guardarArchivos.php". La solicitud se envía como tipo POST y se incluye el objeto FormData como datos. Se configuran las opciones de cache, contenido y procesamiento de datos como falso.



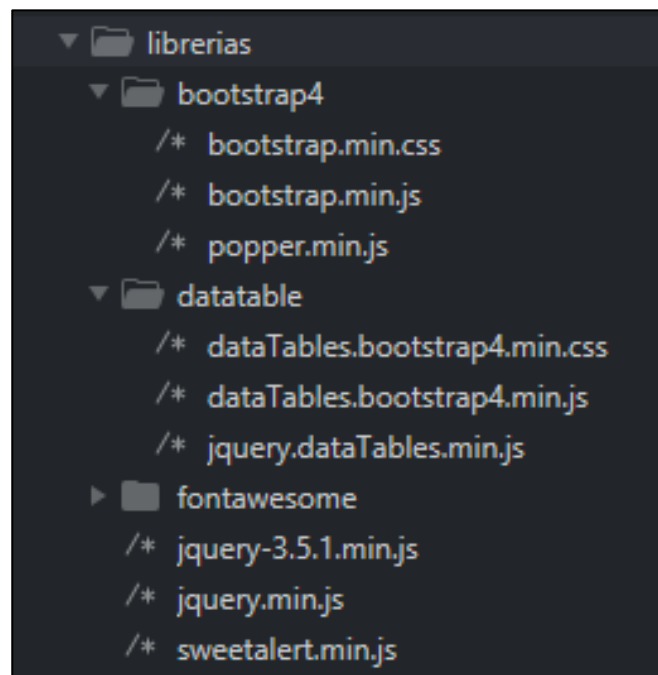
Si la respuesta del servidor es 1, se restablece el formulario, se recarga la tabla de archivos y se muestra una notificación de éxito. De lo contrario, se muestra una notificación de error.

- b) La función "eliminarArchivo(idArchivo)" se utiliza para eliminar un archivo del gestor. Muestra una ventana emergente de confirmación para verificar si el usuario desea eliminar el archivo. Si el usuario confirma la eliminación, realiza una solicitud AJAX al servidor utilizando el archivo "eliminaArchivo.php" y el ID del archivo como datos. Si la respuesta del servidor es 1, se recarga la tabla de archivos y se muestra una notificación de éxito. De lo contrario, se muestra una notificación de error.
- c) La función "obtenerArchivoPorId(idArchivo)" se utiliza para obtener un archivo específico del gestor. Realiza una solicitud AJAX al servidor utilizando el archivo "obtenerArchivo.php" y el ID del archivo como datos. La respuesta del servidor se inserta en el elemento HTML con el ID "archivoObtenido".

En la **Figura 28**, se encuentra la quinta carpeta denominada "librerías", la cual es necesaria para el desarrollo del gestor de archivos.

Figura 28

Librerías para el desarrollo del gestor de archivos





- **Bootstrap 4:**
 - a) **Librería bootstrap.min.css:** Es una hoja de estilos CSS que pertenece al framework Bootstrap. Este archivo se utiliza para aplicar los estilos y la apariencia visual predefinida de Bootstrap a un sitio web, proporciona un conjunto de reglas de estilo CSS listas para usar que permiten diseñar y personalizar fácilmente el aspecto de un sitio web utilizando los componentes y clases de estilo de Bootstrap.
 - b) **Librería bootstrap.min.js:** Es un archivo JavaScript que forma parte del framework Bootstrap. Este archivo se utiliza para agregar funcionalidades interactivas y dinámicas a un sitio web.
 - c) **Librería popper.min.js:** Es una dependencia utilizada en conjunto con el framework Bootstrap. Esta librería es necesaria para manejar el posicionamiento y el comportamiento de los elementos.
- **Data table:**
 - a) **dataTables.bootstrap4.min.css:** Este archivo contiene reglas y estilos predefinidos que permiten darle una apariencia y formato visual específico a las tablas.
 - b) **dataTables.bootstrap4.min.js:** Este archivo contiene código y funciones predefinidas que permiten mejorar la funcionalidad y el comportamiento de las tablas.
 - c) **jquery.dataTables.min.js:** Este archivo contiene el código y las funciones necesarias para proporcionar funcionalidades avanzadas a las tablas HTML, como ordenamiento, paginación, filtrado y búsqueda, mediante el uso de la biblioteca jQuery.
- **Fontawesome:**
 - a) **Jquery-3.5.1.min.js:** Contiene el código de la biblioteca jQuery en su versión 3.5.1. Esta biblioteca proporciona una amplia gama de funciones y utilidades que simplifican la manipulación y el manejo de elementos HTML, eventos, animaciones, comunicación con el servidor y muchas otras tareas comunes en desarrollo web. Realizar tareas comunes de manera más rápida y sencilla.
 - b) **Jquery.min.js:** La biblioteca jQuery.min.js es una versión comprimida y optimizada de la biblioteca jQuery. Aunque no se



especifica una versión específica, generalmente se refiere a la última versión estable disponible.

- c) **sweetalert.min.js:** Es una biblioteca de JavaScript que proporciona una interfaz de usuario atractiva y personalizable para mostrar mensajes de alerta en los navegadores web. El archivo "sweetalert.min.js" es la versión comprimida y minimizada de esta biblioteca, permite mostrar alertas emergentes personalizadas en lugar de las alertas predeterminadas del navegador.

En la **Figura 29**, la sexta carpeta se llama "procesos" y contiene las subcarpetas "categorías," "gestor," y "usuario." En la carpeta "Categorías" se encuentran los siguientes archivos: "actualizarCategoria.php," "agregarCategoria.php," "eliminarCategoria.php," y "obtenerCategoria.php."

Estos códigos utilizan la clase "Categorías" para actualizar, agregar, eliminar y obtener una categoría en la base de datos y devuelve el resultado de la operación.

Figura 29

Clase "Categorías" para actualizar, agregar, eliminar y obtener una categoría en la base de datos.

```
actualizaCategoria.php x
1 <?php
2
3 require_once "../../clases/Categorias.php";
4 $Categorias = new Categorias();
5
6 $datos = array (
7     "idCategoria" => $_POST['idCategoria'],
8     "categoria" => $_POST['categoriaU']
9 );
10
11 echo $Categorias->actualizarCategoria($datos);
12
13 ?>
```

```
agregarCategoria.php x
1 <?php
2
3 session_start();
4
5 require_once "../../clases/Categorias.php";
6 $Categorias = new Categorias();
7
8 $datos = array (
9     "idUsuario" => $_SESSION['idUsuario'],
10    "categoria" => $_POST['categoria']
11 );
12 echo $Categorias->agregarCategoria($datos);
13
14 ?>
```



```
eliminarCategoria.php x
1 <?php
2
3     session_start();
4     require_once "../../clases/Categorias.php";
5     $Categorias = new Categorias();
6
7     echo $Categorias->eliminarCategorias($_POST['idCategoria']);
8
9
10  ?>
```

```
obtenerCategoria.php x
1 <?php
2
3     require_once "../../clases/Categorias.php";
4     $Categorias = new Categorias();
5
6     echo json_encode($Categorias->obtenerCategoria($_POST['idCategoria']));
7  ?>
```

a) Actualizar categoría:

- Se instancia la clase "Categorías".
- Se crea un array con los datos de la categoría a actualizar.
- Se llama al método "actualizarCategoria ()" de la instancia de la clase "Categorías" y se muestra el resultado.

b) Agregar categoría:

- Se inicia la sesión.
- Se instancia la clase "Categorías".
- Se crea un array con los datos de la categoría a agregar.
- Se llama al método "agregarCategoria ()" de la instancia de la clase "Categorías" y se muestra el resultado.

c) Eliminar categoría:

- Se inicia la sesión.
- Se instancia la clase "Categorías".
- Se llama al método "eliminarCategorias ()" de la instancia de la clase "Categorías" y se muestra el resultado.

d) Obtener categoría:

- Se instancia la clase "Categorías".
- Se llama al método "obtenerCategoria ()" de la instancia de la clase

"Categorías" y se muestra el resultado en formato JSON.

Dentro de la carpeta Gestor están los archivos: eliminarArchivo.php, guardarArchivos.php y obtenerArchivo.php.

- **EliminaArchivo.php:** En la **Figura 30**, este código elimina un archivo físico en el servidor y elimina el registro correspondiente en la base de datos utilizando la clase "Gestor".

Figura 30

Eliminar un archivo del servidor y eliminar el registro en la base de datos

```
eliminaArchivo.php
1 <?php
2
3 session_start();
4 require_once "../../clases/Gestor.php";
5 $Gestor = new Gestor();
6 $idArchivo = $_POST['idArchivo'];
7 $idUserario = $_SESSION['idUserario'];
8
9 $nombreArchivo = $Gestor->obtenNombreAchivo($idArchivo);
10
11 $rutaEliminar = "../../archivos/" . $idUserario . "/" . $nombreArchivo;
12
13 if (unlink($rutaEliminar)) {
14 echo $Gestor->eliminaRegistroArchivo($idArchivo);
15 } else {
16     echo 0;
17 }
18
19
20
21 ?>
```

- a) Inicia la sesión utilizando el comando "session_start()". Esto permite acceder a las variables de sesión.
- b) Se incluye el archivo que contiene la definición de la clase "Gestor".
- c) Se crea una instancia de la clase "Gestor".
- d) Se obtiene el valor del ID de archivo desde la solicitud POST y se asigna a la variable \$idArchivo.
- e) Se obtiene el ID de usuario desde la variable de sesión \$_SESSION y se asigna a la variable \$idUserario.
- f) Se llama al método "obtenNombreAchivo()" de la instancia de la clase "Gestor" para obtener el nombre del archivo correspondiente al ID proporcionado.
- g) Se construye la ruta del archivo a eliminar concatenando el directorio de archivos, el ID de usuario y el nombre del archivo.

- h) Se utiliza la función "unlink()" para eliminar físicamente el archivo de la ruta especificada. Si la eliminación es exitosa, se llama al método "eliminaRegistroArchivo()" de la instancia de la clase "Gestor" y se muestra el resultado. En caso contrario, se muestra el valor 0.
- **GuardarArchivos.php:** En la **Figura 31**, este código permite subir múltiples archivos al servidor, guardarlos en una carpeta asignada al usuario y registrarlos en la base de datos mediante la utilización de la clase "Gestor." Posteriormente, muestra el resultado de la operación en pantalla.

Figura 31

Subir Múltiples archivos al servidor y registrarlos

```
guardarArchivos.php x
1 <?php
2 session_start();
3 require_once "../clases/Gestor.php";
4 $Gestor = new Gestor();
5 $idCategoria = $_POST['categoriasArchivos'];
6 $idUserario = $_SESSION['idUserario'];
7
8 if ($_FILES['archivos']['size'] > 0) {
9
10     $carpetaUsuario = '../archivos/'.$idUserario;
11
12     if (!file_exists($carpetaUsuario)){
13         mkdir($carpetaUsuario, 0777, true);
14     }
15
16     $totalArchivos = count($_FILES['archivos']['name']);
17     for ($i=0; $i < $totalArchivos; $i++) {
18
19         $nombreArchivo = $_FILES['archivos']['name'][$i];
20         $explode = explode('.', $nombreArchivo);
21         $tipoArchivo = array_pop($explode);
22         $rutaAlmacenamiento = $_FILES['archivos']['tmp_name'][$i];
23         $rutaFinal = $carpetaUsuario ."/". $nombreArchivo;
24
25         $datosRegistroArchivo = array(
26             "idUserario" => $idUserario,
27             "idCategoria" => $idCategoria,
28             "nombreArchivo" => $nombreArchivo,
29             "tipo" => $tipoArchivo,
30             "ruta" => $rutaFinal
31             );
32
33         if (move_uploaded_file($rutaAlmacenamiento, $rutaFinal)) {
34             $respuesta = $Gestor->agregaRegistroArchivo($datosRegistroArchivo);
35         }
36         echo $respuesta;
37     } else {
38         echo 0;
39     }
40 }
41 >>
```

- a) Inicia la sesión utilizando el comando "session_start()". Esto permite acceder a las variables de sesión.
- b) Se incluye el archivo que contiene la definición de la clase "Gestor".
- c) Se crea una instancia de la clase "Gestor".
- d) Se obtiene el valor del ID de categoría desde la solicitud POST y se asigna a la variable \$idCategoria.



- e) Se obtiene el ID de usuario desde la variable de sesión `$_SESSION` y se asigna a la variable `$idUsuario`.
- f) Se verifica si se han enviado archivos a través de la solicitud POST.
- g) Si se han enviado archivos:
 - Se crea una carpeta de usuario en el directorio "archivos" si no existe.
 - Se obtiene el número total de archivos enviados.
 - Se itera sobre cada archivo enviado y se realizan las siguientes acciones:
 - Se obtiene el nombre del archivo, su extensión y la ruta temporal de almacenamiento.
 - Se construye la ruta final de almacenamiento concatenando la carpeta de usuario y el nombre del archivo.
 - Se crea un array con los datos del archivo para su registro en la base de datos.
 - Si se mueve el archivo a la ruta final de almacenamiento, se llama al método "agregaRegistroArchivo ()" de la instancia de la clase "Gestor" para agregar el registro en la base de datos.
- h) Se muestra el resultado de la operación (1 para éxito o 0 para fallo) mediante el comando "echo".
- **ObtenerArchivo.php:** En la **Figura 32**, este código obtiene la ruta de un archivo en el servidor utilizando la clase "Gestor" y la muestra en pantalla.

Figura 32

Ruta de archivo en el servidor

```
obtenerArchivo.php x
1  <?php
2
3      session_start();
4      require_once "../..//clases/Gestor.php";
5      $Gestor = new Gestor();
6      $idArchivo = $_POST['idArchivo'];
7
8      echo $Gestor->obtenerRutaArchivo($idArchivo);
9
10  ?>
```



Dentro de la carpeta Usuario hay dos sub carpetas: login, registro y un archivo salir.php.

En la sub carpeta login está el archivo "login.php"

- **Login.php:** En la **Figura 33**, este código lleva a cabo el proceso de inicio de sesión de un usuario utilizando la clase "Usuario" y muestra el resultado en pantalla.

Figura 33

Proceso de inicio de sesión de usuario.

```
login.php x
1  <?php
2  session_start();
3  require_once "../../clases/Usuario.php";
4
5  $usuario = $_POST['login'];
6  $password = sha1($_POST['password']);
7
8  $usuarioObj = new Usuario();
9
10 echo $usuarioObj->login($usuario, $password);
11
12 ?>
```

- a) Inicia la sesión utilizando el comando "session_start()". Esto permite acceder a las variables de sesión.
- b) Se incluye el archivo que contiene la definición de la clase "Usuario".
- c) Se obtiene el valor del nombre de usuario desde la solicitud POST y se asigna a la variable \$usuario.
- d) Se obtiene el valor de la contraseña desde la solicitud POST y se encripta utilizando la función "sha1()", y se asigna a la variable \$password.
- e) Se crea una instancia de la clase "Usuario".
- f) Se llama al método "login()" de la instancia de la clase "Usuario", pasando como parámetros el nombre de usuario y la contraseña encriptada.
- g) Se muestra el resultado del inicio de sesión (1 para éxito o 0 para fallo) mediante el comando "echo".

En la sub carpeta registro está el archivo "agregarUsuario.php"

- **agregarUsuario.php:** En la **Figura 34**, este código agrega un nuevo usuario utilizando la clase "Usuario" y pasa los datos del usuario obtenidos a través de la solicitud POST.

Figura 34

Agrega un nuevo usuario utilizando la clase "Usuario"

```
agregarUsuario.php x
1  <?php
2
3      require_once "../.../clases/Usuario.php";
4
5      $password = sha1($_POST['password']);
6      $datos = array(
7          "nombre" => $_POST['nombre'],
8          "apellidos" => $_POST['apellidos'],
9          "dni" => $_POST['dni'],
10         "email" => $_POST['email'],
11         "usuario" => $_POST['usuario'],
12         "password" => $password
13     );
14     $usuario = new Usuario();
15     echo $usuario->agregarUsuario($datos);
16
17  ?>
```

- a) Se incluye el archivo que contiene la definición de la clase "Usuario".
- b) Se encripta el valor de la contraseña desde la solicitud POST utilizando la función "sha1()", y se asigna a la variable \$password.
- c) Se crea un array llamado \$datos que contiene los valores de los campos nombre, apellidos, DNI, email, usuario y contraseña encriptada, obtenidos desde la solicitud POST.
- d) Se crea una instancia de la clase "Usuario".
- e) Se llama al método "agregarUsuario()" de la instancia de la clase "Usuario", pasando como parámetro el array \$datos.

- **Salir.php:** En la **Figura 35**, este código cierra la sesión del usuario y lo redirecciona a la página de inicio.

Figura 35

Cierra la sesión del usuario y lo redirecciona al inicio.

```
salir.php x
1  <?php
2
3      session_start();
4      session_destroy();
5
6      header("location:../..//index.php");
7
8  ?>
```



- a) Inicia la sesión utilizando el comando "session_start()". Esto permite acceder a las variables de sesión.
- b) Destruye la sesión utilizando el comando "session_destroy()". Esto elimina todas las variables de sesión y finaliza la sesión actual.
- c) Redirecciona al usuario a la página de inicio (index.php) utilizando la función "header()".
- d) Al redireccionar, el usuario será dirigido a la página index.php.

Como séptima carpeta se tiene el archivo "vistas" y como sub carpetas están: categorías, gestor.

- Subcarpeta " categorías ":
- **Archivo "selectCategorias.php"**: En la **Figura 36**, este código obtiene las categorías vinculadas a un usuario específico y las presenta como opciones en un elemento "select" en HTML. Esto habilita al usuario para elegir una categoría con la que interactuar en el sistema.

Figura 36

Muestra las categorías asociadas a un usuario específico

```
selectCategorias.php
1  <?php
2
3      session_start();
4      require_once "../../clases/Conexion.php";
5      $c = new Conectar();
6      $conexion = $c->conexion();
7
8      $idUserario = $_SESSION['idUserario'];
9      $sql = "SELECT id_categoria,
10             nombre
11             FROM t_categorias
12             WHERE id_usuario = '$idUserario'";
13      $result = mysqli_query($conexion, $sql);
14
15  ?>
16
17  <select name="categoriasArchivos" id="
18  categoriasArchivos" class="form-control">
19  <?php
20      while($mostrar = mysqli_fetch_array($
21      result)){
22          $idCategoria = $mostrar['id_categoria']
23          ;
24      }
25  ?>
26  </select>
```

- a) Inicia la sesión utilizando el comando "session_start()". Esto permite acceder a las variables de sesión.



- b) Importa el archivo "Conexion.php" que contiene la clase "Conectar".
 - c) Crea una instancia de la clase "Conectar" llamada "\$c".
 - d) Establece la conexión a la base de datos utilizando el método "conexion()" de la clase "Conectar".
 - e) Obtiene el ID de usuario almacenado en la variable de sesión "\$_SESSION['idUsuario']".
 - f) Realiza una consulta SQL para obtener las categorías asociadas al ID de usuario en la tabla "t_categorias".
 - g) Ejecuta la consulta utilizando la función "mysqli_query()" pasando la conexión y la consulta SQL.
 - h) Muestra las opciones de un elemento "select" en HTML con los resultados obtenidos de la consulta.
 - i) Dentro de un bucle "while", recorre cada fila de resultados y genera una opción para el elemento "select" con el ID de la categoría y su nombre.
 - j) Cierra el bucle y el elemento "select".
- **Archivo "tablaCategoria.php":** En la **Figura 37**, este código muestra una tabla HTML con las categorías relacionadas a un usuario en particular. Ofrece funcionalidades de edición y eliminación de categorías, aprovechando DataTables para mejorar la experiencia del usuario con características como ordenamiento y paginación en la tabla.

Figura 37

Tabla HTML con las categorías asociadas al usuario específico.

```
tablaCategoria.php
2 <?php
3     session_start();
4     require_once "../clases/Conexion.php";
5     $idUsuario = $_SESSION['idUsuario'];
6     $conexion = new Conectar();
7     $conexion = $conexion->conexion();
8
9     ?>
10 <div class="table-reponsive">
11 <table class="table table-hover table-dark
12 table-bordered" id="
13 tablaCategoriasDataTable">
14 <thead>
15 <tr style="text-align: center;">
16 <td>Nombre</td>
17 <td>Fecha</td>
18 <td>Editar</td>
19 <td>Eliminar</td>
20 </tr>
21 </thead>
22 <tbody>
```



```
22 <?php
23     $sql = "SELECT id_categoria,
24               nombre,
25               fechaInsert
26               FROM t_categorias
27               WHERE id_usuario = '$idUsuario';
28     $result = mysqli_query($conexion, $sql);
29
30     while($mostrar = mysqli_fetch_array($result)){
31         $idCategoria = $mostrar['id_categoria'];
32     }
33
34     <tr style="text-align: center;">
35         <td><?php echo $mostrar['nombre']; ?></td>
36         <td><?php echo $mostrar['fechaInsert']; ?></td>
37         <td>
38             <span class="btn btn-warning btn-sm"
39                 onclick="obtenerDatosCategoria('<?php echo $idCategoria ?>')"
40                 data-toggle="modal" data-target="#modalActualizarCategoria">
41                 <span class="fas fa-edit"></span>
42             </span>
43         </td>
44         <td>
45             <span class="btn btn-danger btn-sm"
46                 onclick="eliminarCategorias('<?php echo $idCategoria ?>')">
47                 <span class="fas fa-trash-alt"></span>
48             </span>
49         </td>
50     </tr>
51 <?php
52     }
53 <?>
54 </tbody>
55 </table>
56 </div>
57 <script type="text/javascript">
58     $(document).ready(function(){
59         $('#tablaCategoriasDataTable').DataTable();
60     });
61 </script>
```

- a) Inicia la sesión utilizando el comando "session_start()". Esto permite acceder a las variables de sesión.
- b) Importa el archivo "Conexion.php" que contiene la clase "Conectar".
- c) Obtiene el ID de usuario almacenado en la variable de sesión "\$_SESSION['idUsuario']".
- d) Crea una instancia de la clase "Conectar" llamada "\$conexion" y establece la conexión a la base de datos utilizando el método "conexion()".
- e) Crea una tabla HTML con la clase "table" y otros atributos para el formato y estilo.
- f) Dentro de la tabla, crea un encabezado con las columnas "Nombre", "Fecha", "Editar" y "Eliminar".
- g) Realiza una consulta SQL para obtener las categorías asociadas al ID de usuario en la tabla "t_categorias".
- h) Ejecuta la consulta utilizando la función "mysqli_query()" pasando la conexión y la consulta SQL.



- i) Dentro de un bucle "while", recorre cada fila de resultados y genera una fila en la tabla con el nombre de la categoría, la fecha de inserción y botones para editar y eliminar.
- j) Cierra el bucle y la tabla.
- k) Agrega un script JavaScript que utiliza la biblioteca DataTables para aplicar la funcionalidad de ordenamiento y paginación a la tabla.

- **Subcarpeta "gestor":**
- **Archivo "tablaGestor.php":** En la **Figura 38**, este código presenta una tabla HTML con los detalles de los archivos guardados por el usuario actual. Ofrece funcionalidad para descargar, visualizar y eliminar archivos, y se beneficia de DataTables para mejorar la experiencia del usuario con características como ordenamiento y paginación en la tabla.

Figura 38

Funcionalidades para descargar, visualizar y eliminar archivos que almacena el usuario.

```
tablaGestor.php
1 <?php
2 session_start();
3 require_once "../clases/Conexion.php";
4 $c = new conectar();
5 $conexion = $c->conexion();
6 $idUserario = $_SESSION['idUserario'];
7 $sql = "SELECT
8     archivos.id_archivo as idArchivo,
9     usuario.nombre as nombreUsuario,
10    categorias.nombre as categoria,
11    archivos.nombre as nombreArchivo,
12    archivos.tipo as tipoArchivo,
13    archivos.ruta as rutaArchivo,
14    archivos.fecha as fecha
15 FROM
16     t_archivos AS archivos
17     INNER JOIN
18     t_usuarios AS usuario ON archivos.id_usuario = usuario.idt_usuario
19     INNER JOIN
20     t_categorias AS categorias ON archivos.id_categoria = categorias.id_categoria
21     and archivos.id_usuario = " . $idUserario . "'";
22 $result = mysqli_query($conexion, $sql);
23
24
25 ?>
26
27
28 <div class="erow">
29     <div class="col-sm-12">
30         <div class="table-responsive">
31             <table class="table table-hover table-dark table-bordered" id="tablaGestorDataTable">
32                 <thead>
33                     <tr>
34                         <th>Categoria</th>
35                         <th>Nombre</th>
36                         <th>Extensión archivo</th>
37                         <th>Descargar</th>
38                         <th>Visualizar</th>
39                         <th>Eliminar</th>
40                     </tr>
41                 </thead>
42                 <tbody>
43                 <?php
44
45                     /*
46                     Arreglo de extensiones validas
47                     */
48
49                     $extensionesValidas = array('png', 'jpg', 'pdf', 'mp4', 'mp3');
```



```
51 while($mostrar = mysqli_fetch_array($result)) {
52
53     $rutaDescarga = "../archivos/".$idUsuario."/".$mostrar['nombreArchivo'];
54     $nombreArchivo = $mostrar['nombreArchivo'];
55     $idArchivo = $mostrar['idArchivo'];
56 }>
57 <tr>
58 <td><?php echo $mostrar['categoria']; ?></td>
59 <td><?php echo $mostrar['nombreArchivo']; ?></td>
60 <td><?php echo $mostrar['tipoArchivo']; ?></td>
61 <td>
62 <a href="<?php echo $rutaDescarga; ?>"
63     download="<?php echo $nombreArchivo; ?>" class="btn btn-warning btn-sm">
64     <span class="fas fa-download"></span>
65 </a>
66 </td>
67 <td>
68 <?php
69 for ($i=0; $i < count($extensionesValidas); $i++) {
70     if ($extensionesValidas[$i] == $mostrar['tipoArchivo']) {
71     }>
72 <span class="btn btn-primary btn-sm"
73     data-toggle="modal"
74     data-target="#visualizarArchivo"
75     onclick="obtenerArchivoPorId('<?php echo $idArchivo ?>')">
76 <span class="fas fa-eye"></span>
77 </span>
78 <?php
79     }
80 }>
81 </td>
82 <td>
83 <span class="btn btn-danger btn-sm"
84     onclick="eliminarArchivo('<?php echo $idArchivo ?>')">
85 <span class="fas fa-trash-alt"></span>
86 </span>
87 </td>
88 </tr>
89 </tbody>
90 <?php
91 }
92 >
93 </tbody>
94 </table>
95 </div>
96 </div>
97 </div>
98
```

- Inicia la sesión utilizando el comando "session_start()". Esto permite acceder a las variables de sesión.
- Importa el archivo "Conexion.php" que contiene la clase "Conectar".
- Crea una instancia de la clase "Conectar" llamada "\$c" y establece la conexión a la base de datos utilizando el método "conexion()".
- Obtiene el ID de usuario almacenado en la variable de sesión "\$_SESSION['idUsuario']".
- Crea una consulta SQL para obtener los detalles de los archivos almacenados en la tabla "t_archivos" que pertenecen al usuario actual.
- Ejecuta la consulta utilizando la función "mysqli_query()" pasando la conexión y la consulta SQL.
- Crea una tabla HTML con la clase "table" y otros atributos para el formato y estilo.
- Dentro de la tabla, crea un encabezado con las columnas "Categoría", "Nombre", "Extensión archivo", "Descargar", "Visualizar" y "Eliminar".



- i) Crea un bucle "while" que recorre cada fila de resultados y genera una fila en la tabla con la categoría, el nombre del archivo, la extensión del archivo y botones para descargar, visualizar y eliminar.
 - j) Verifica si la extensión del archivo es una de las extensiones válidas definidas en el arreglo "\$extensionesValidas". Si es válida, muestra el botón para visualizar el archivo.
 - k) Cierra el bucle y la tabla.
 - l) Agrega un script JavaScript que utiliza la biblioteca DataTables para aplicar la funcionalidad de ordenamiento y paginación a la tabla.
- **Archivo "categorías.php":** En la **Figura 39**, este código exhibe una página de gestión de categorías que permite agregar, actualizar y eliminar categorías. Además, cuenta con funcionalidades JavaScript y depende de archivos de estilo y scripts adicionales. Además, verifica si el usuario ha iniciado sesión, y en caso afirmativo, incluye los archivos "header.php" y "footer.php".

Figura 39

Página de administración de categorías (agregar, actualizar y eliminar categorías)

```
categorias.php
4     if(isset($_SESSION['usuario'])){
5         include "header.php";
6     }
7
8     <div class="jumbotron jumbotron-fluid">
9         <div class="container">
10            <h1 class="display-4">Categoría</h1>
11
12            <div class="row">
13                <div class="col-sm-4">
14                    <span class="btn btn-success" data-toggle="modal" data-target="#modalAgregaCategoria">
15                        <span class="fas fa-plus-circle"></span> Agregar nueva categoría
16                    </span>
17                </div>
18            </div>
19            <hr>
20            <div class="row">
21                <div class="col-sm-12">
22                    <div id="tablaCategorias">
23                        <div>
24                        </div>
25                    </div>
26                </div>
27
28            <!-- Modal -->
29            <div class="modal fade" id="modalAgregaCategoria" tabindex="-1" role="dialog" aria-labelledby="exampleModallabel" aria-hidden="true">
30                <div class="modal-dialog modal-sm" role="document">
31                    <div class="modal-content">
32                        <div class="modal-header">
33                            <h5 class="modal-title" id="exampleModallabel">Agregar nueva categoría</h5>
34                            <button type="button" class="close" data-dismiss="modal" aria-label="Close">
35                                <span aria-hidden="true">&times;</span>
36                            </button>
37                        </div>
38                        <div class="modal-body">
39                            <div class="form">
40                                <input type="text" name="nombreCategoria" id="nombreCategoria" class="form-control">
41                            </div>
42                        </div>
43                        <div class="modal-footer">
44                            <button type="button" class="btn btn-secondary" data-dismiss="modal">Cerrar</button>
45                            <button type="button" class="btn btn-primary" id="btnGuardarCategoria">Guardar</button>
46                        </div>
47                    </div>
48                </div>
49            </div>
50
51            <!-- Modal -->
52            <div class="modal fade" id="modalActualizarCategoria" tabindex="-1" role="dialog" aria-labelledby="exampleModallabel" aria-hidden="true">
53                <div class="modal-dialog modal-sm" role="document">
54                    <div class="modal-content">
55                        <div class="modal-header">
56                            <h5 class="modal-title" id="exampleModallabel">Actualizar categoría</h5>
57                            <button type="button" class="close" data-dismiss="modal" aria-label="Close">
58                                <span aria-hidden="true">&times;</span>
59                            </button>
60                        </div>
61                        <div class="modal-body">
62                            <div class="form">
63                                <input type="text" id="idCategoria" name="idCategoria" hidden="">
64                                <input type="text" id="categoriaU" name="categoriaU" class="form-control">
65                            </div>
66                        </div>
67                        <div class="modal-footer">
68                            <button type="button" class="btn btn-secondary" data-dismiss="modal">Cerrar</button>
69                            <button type="button" class="btn btn-warning" id="btnActualizarCategoria">Actualizar</button>
70                        </div>
71                    </div>
72                </div>
73            </div>
74
75            <!-- Modal -->
76            <div class="modal fade" id="modalEliminarCategoria" tabindex="-1" role="dialog" aria-labelledby="exampleModallabel" aria-hidden="true">
77                <div class="modal-dialog modal-sm" role="document">
78                    <div class="modal-content">
79                        <div class="modal-header">
80                            <h5 class="modal-title" id="exampleModallabel">Eliminar categoría</h5>
81                            <button type="button" class="close" data-dismiss="modal" aria-label="Close">
82                                <span aria-hidden="true">&times;</span>
83                            </button>
84                        </div>
85                        <div class="modal-body">
86                            <div class="form">
87                                <input type="text" id="idCategoria" name="idCategoria" hidden="">
88                            </div>
89                        </div>
90                        <div class="modal-footer">
91                            <button type="button" class="btn btn-secondary" data-dismiss="modal">Cerrar</button>
92                            <button type="button" class="btn btn-danger" id="btnEliminarCategoria">Eliminar</button>
93                        </div>
94                    </div>
95                </div>
96            </div>
97        </div>
98    </div>
```

Fig. 71



```
85 <!--Funciones js de Categorías, las dependencias-->
86 <script src="../../js/categorias.js"></script>
87
88 <script type="text/javascript">
89 $(document).ready(function(){
90     $('#tablaCategorias').load("categorias/tablaCategoria.php");
91
92     $('#btnGuardarCategoria').click(function(){
93         agregarCategoria();
94     });
95
96     $('#btnActualizarCategoria').click(function(){
97         actualizarCategoria();
98     });
99 });
100 </script>
101
102 <?php
103 }else {
104     header("location:../index.php");
105 }
106 ?>
107
```

El contenido de la página incluye:

- Un encabezado "Categoría" dentro de un jumbotron.
- Un botón "Agregar nueva categoría" que abre un modal para ingresar el nombre de la nueva categoría.
- Un contenedor "tablaCategorias" que se actualizará con la tabla de categorías.
- Dos modales: "modalAgregaCategoria" para agregar una nueva categoría y "modalActualizarCategoria" para actualizar una categoría existente.
- Se incluye el archivo "categorias.js" que contiene las funciones JavaScript relacionadas con la gestión de categorías.

En la parte final del código, se cargan los datos de la tabla de categorías al cargar la página y se asignan las funciones correspondientes a los botones "Guardar" y "Actualizar".

- Archivo "footer.php":** En la **Figura 40**, se incluyen varios scripts de librerías en un archivo HTML. Estos scripts están diseñados para mejorar la experiencia del usuario y ofrecer funcionalidades adicionales en el sitio web.

Figura 40

Scripts de librerías en un archivo HTML

```
footer.php
1 <script src="../../librerias/jquery-3.5.1.min.js"></script>
2 <script src="../../librerias/bootstrap4/popper.min.js"></script>
3 <script src="../../librerias/bootstrap4/bootstrap.min.js"></script>
4 <script src="../../librerias/sweetalert.min.js"></script>
5 <script src="../../librerias/datatables/jquery.dataTables.min.js"></script>
6 <script src="../../librerias/datatables/dataTables.bootstrap4.min.js"></script>
7 </body>
8
9 </html>
```

Se hace mención de cada una en la carpeta quinta que contiene el archivo "librerías".

- **Archivo "gestor.php"**: En la **Figura 41**, se presenta una página web de un gestor de archivos. Si la sesión del usuario está activa, se incorpora el encabezado de la página. La página se compone de un encabezado amplio con el título "Gestor de archivos Gremh", seguido de un botón "Agregar archivos" que despliega un modal. A continuación, se encuentra un contenedor destinado a mostrar la tabla de archivos del gestor.

Figura 41

Interfaz del gestor de archivos y proporciona funcionalidad para agregar y visualizar archivos.

```
gestor.php
1 <?php
2 session_start();
3 if(isset($_SESSION['usuario'])){
4
5
6 include "header.php";
7
8 }>
9
10 <div class="jumbotron jumbotron-fluid">
11 <div class="container">
12 <h1 class="display-4">Gestor de archivos Gremh</h1>
13 <span class="btn btn-success" data-toggle="modal" data-target="#modalAgregarArchivos">
14 <span class="fas fa-plus-circle"></span> Agregar archivos
15 </span>
16 <hr>
17 <div id="tablaGestorArchivos"></div>
18 </div>
19 </div>
20
21 <!-- Modal para agregar Archivos Gremh-->
22
23
24 <!-- Modal -->
25 <div class="modal fade" id="modalAgregarArchivos" tabindex="-1" role="dialog" aria-labelledby="exampleModallabel" aria-hidden="true">
26 <div class="modal-dialog" role="document">
27 <div class="modal-content">
28 <div class="modal-header">
29 <h5 class="modal-title" id="exampleModallabel">Modal title</h5>
30 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
31 <span aria-hidden="true">&times;</span>
32 </button>
33 </div>
34 <div class="modal-body">
35 <form id="frmArchivos" enctype="multipart/form-data" method="post">
36 <label class="form-control-label">Categoria</label>
37 <div id="categoriasLoad"></div>
38 <label class="form-control-label">Selecciona archivos</label>
39 <input type="file" name="archivos[]" id="archivos" class="form-control" multiple>
40 </form>
41 </div>
42 <div class="modal-footer">
43 <button type="button" class="btn btn-secondary" data-dismiss="modal">Cerrar</button>
44 <button type="button" class="btn btn-primary" id="btnGuardarArchivos">Guardar</button>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50
51
```

```
52 <!-- Modal -->
53 <div class="modal fade" id="visualizarArchivo" tabindex="-1" role="dialog" aria-labelledby="exampleModallabel" aria-hidden="true">
54 <div class="modal-dialog modal-lg" role="document">
55 <div class="modal-content">
56 <div class="modal-header">
57 <h5 class="modal-title" id="exampleModallabel">Archivo</h5>
58 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
59 <span aria-hidden="true">&times;</span>
60 </button>
61 </div>
62 <div class="modal-body">
63 <div id="archivoObtenido"></div>
64 </div>
65 <div class="modal-footer">
66 <button type="button" class="btn btn-secondary" data-dismiss="modal">Cerrar</button>
67 </div>
68 </div>
69 </div>
70 </div>
71
72 <?php include "footer.php"; ?>
73
74 <script src="./js/gestor.js"></script>
75 <script type="text/javascript">
76 $(document).ready(function(){
77 $("#tablaGestorArchivos").load("gestor/tablaGestor.php");
78 $("#categoriasLoad").load("categorias/selectCategorias.php");
79
80 $("#btnGuardarArchivos").click(function(){
81 agregarArchivosGestor();
82 });
83 });
84 </script>
85
86 <?php
87
88 }else {
89 header("location:../index.php");
90 }
91 ?>
```



El código también incluye dos modales: uno para agregar archivos y otro para visualizar un archivo seleccionado. Ambos modales contienen formularios y elementos interactivos.

Al final del archivo, se incluye el pie de página y se cargan los archivos JavaScript necesarios, como "gestor.js". Estos archivos contienen funciones relacionadas con la interacción del usuario en el gestor de archivos.

Si la sesión del usuario no está activa, se redirecciona al usuario a la página de inicio de sesión.

- **Archivo "header.php"**: En la **Figura 42**, se ilustra la estructura fundamental de una página web, que abarca elementos como el encabezado del documento, la etiqueta <title> para definir el título de la página, la incorporación de archivos CSS, y la especificación del esquema de caracteres y la configuración de la vista.

Figura 42

Estructura de una página web y establece una barra de navegación con enlaces a las diferentes secciones del sitio

```
header.php
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Gestor</title>
5 <link rel="stylesheet" type="text/css" href="../librerias/bootstrap4/bootstrap.min.css">
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
8 <link rel="stylesheet" type="text/css" href="../librerias/fontawesome/css/all.css">
9 <link rel="stylesheet" type="text/css" href="../librerias/datatables/dataTables.bootstrap4.min.css">
10
11 </head>
12 <body style="background-color: #e9ecef">
13
14 <!-- Navigation -->
15 <nav class="navbar navbar-expand-lg navbar-dark bg-dark static-top">
16 <div class="container">
17 <a class="navbar-brand" href="inicio.php">
18 
19 </a>
20 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive"
21 aria-expanded="false" aria-label="Toggle navigation">
22 <span class="navbar-toggler-icon"></span>
23 </button>
24 <div class="collapse navbar-collapse" id="navbarResponsive">
25 <ul class="navbar-nav ml-auto">
26 <li class="nav-item active">
27 <a class="nav-link" href="inicio.php"><span class="fas fa-home"></span> Inicio
28 <span class="sr-only">(current)</span>
29 </a>
30 </li>
31 <li class="nav-item">
32 <a class="nav-link" href="categorias.php"><span class="fas fa-file"></span> Categorías</a>
33 </li>
34 <li class="nav-item">
35 <a class="nav-link" href="gestor.php"><span class="fas fa-folder"></span> Archivos</a>
36 </li>
37 <li class="nav-item">
38 <a class="nav-link" href="../procesos/usuario/salir.php" style="color: yellow">
39 <span class="fas fa-power-off"></span> Salir
40 </a>
41 </li>
42 </ul>
43 </div>
44 </div>
45 </nav>
```

El contenido principal de la página está dentro del elemento <body>. Se presenta una barra de navegación con un logotipo y enlaces a diferentes secciones del sitio.



Los enlaces incluyen iconos de FontAwesome para representar visualmente cada sección.

- **Archivo "inicio.php":** En la **Figura 43**, este código verifica si existe una sesión de usuario activa. En caso afirmativo, se integra el archivo "header.php" para presentar la cabecera de la página.

Figura 43

Verifica la sesión de usuario y muestra una página de bienvenida si el usuario ha iniciado sesión

```
1 <?php
2 session_start();
3
4     if(isset($_SESSION['usuario'])){
5         include "header.php";
6     }
7
8     <div class="container" style="background-image: url('../img/mineria.jpg'); background-repeat: no-repeat; background-size: cover;">
9     <div class="row">
10        <div class="col-sm-12">
11            <div style="background-color: transparent; text-align: center; padding: 20px;">
12                <div style="background-color: rgba(0, 0, 0, 0.5); text-align: center; padding: 20px;">
13                    <h1 style="font-family: 'Arial Black', sans-serif; font-size: 40px; font-weight: bold; color: #fff;">BIENVENIDOS AL GESTOR DE ARCHIVOS
14                    DE LA GREMH</h1>
15                    
16                </div>
17            </div>
18        </div>
19    </div>
20 </div>x
21
22
23 <?php
24 include "footer.php";
25 }else {
26     header("location:../index.php");
27 }
28 ?>
```

Dentro del elemento <div class="container">, se crea una sección de contenido con una imagen de fondo y un mensaje de bienvenida centrado. Se utiliza estilos CSS para ajustar el fondo y el diseño del mensaje.

Finalmente, se incluye el archivo "footer.php" para mostrar el pie de página.

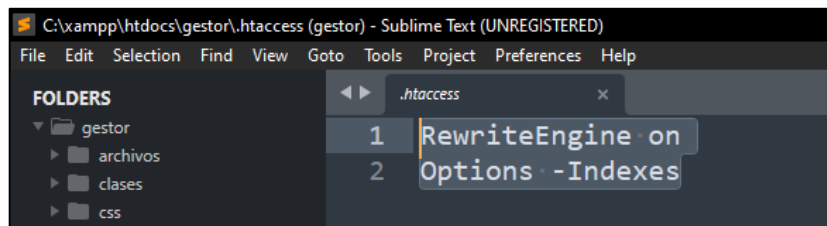
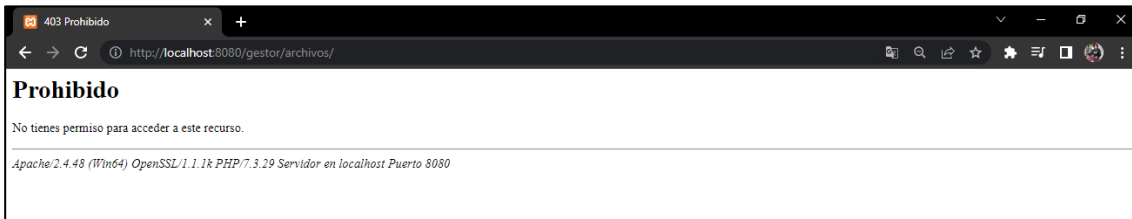
Si no hay una sesión de usuario iniciada, se redirige al usuario a la página de inicio de sesión.

Así mismo se agrega los últimos archivos dentro de la estructura del desarrollo que son: htaccess, index.php y registro.php.

- **Htaccess:** En la **Figura 44**, se establece una medida de seguridad para evitar el acceso a las carpetas de otros usuarios mediante el uso de una regla en el archivo .htaccess. Este archivo es una configuración utilizado por Apache para gestionar aspectos relacionados con la solicitud y entrega de archivos, asegurando la privacidad de los usuarios.

Figura 44

Deshabilitar la opción de ver el listado de archivos

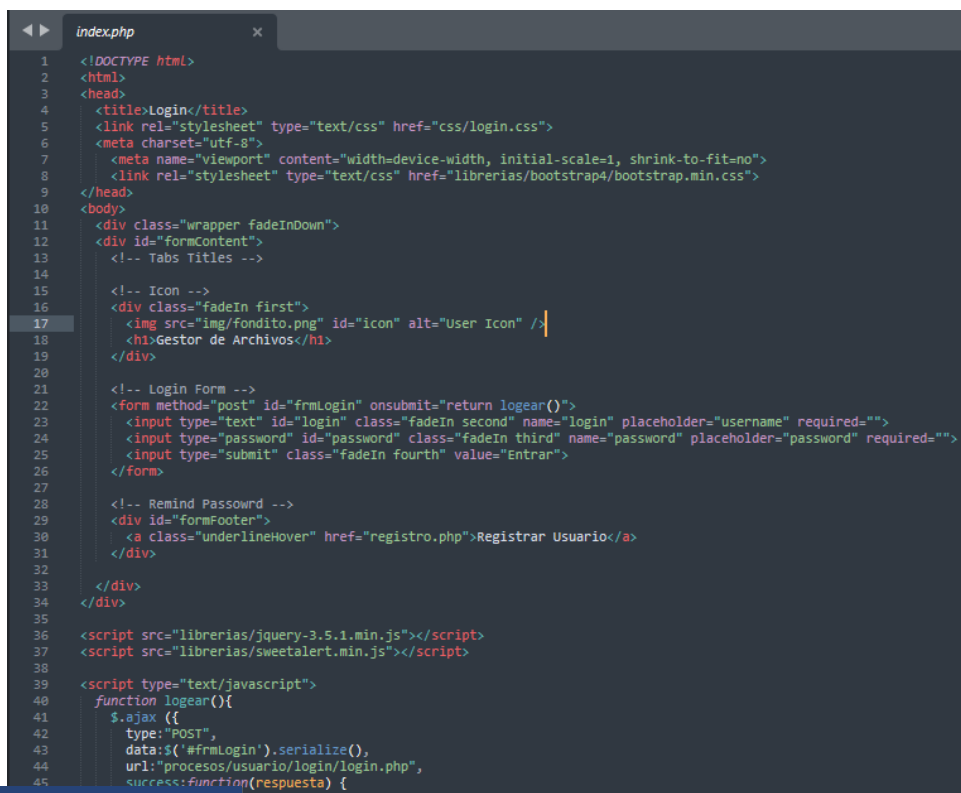


La regla mencionada habilita el motor de reescritura de URL y deshabilita la opción de ver el listado de archivos en un directorio.

- **Index.php:** En la **Figura 45**, este código representa una página de inicio de sesión que consta de un formulario para introducir las credenciales de usuario. También incluye una función JavaScript para validar las credenciales y redirigir al usuario en caso de éxito. El archivo contiene la página de inicio de sesión, junto con estilos CSS, y las librerías Bootstrap y jQuery para mejorar la apariencia y la funcionalidad de la página.

Figura 45

Página de inicio de sesión





```
47     respuesta = respuesta.trim();
48     if (respuesta == 1){
49         window.location = "vistas/inicio.php";
50     } else {
51         swal(":", "Fallo al entrar!", "error");
52     }
53 }
54 });
55 return false;
56 }
57 </script>
58 </body>
59 </html>
```

Dentro del elemento `<div class="wrapper fadeInDown">` se encuentra el formulario de inicio de sesión. El usuario debe ingresar un nombre de usuario y contraseña.

Al enviar el formulario, se realiza una solicitud AJAX a través de jQuery al archivo "login.php" en la carpeta "procesos/usuario/login" para verificar las credenciales ingresadas. Si las credenciales son correctas, se redirige al usuario a la página de inicio, de lo contrario, se muestra un mensaje de error utilizando la librería SweetAlert.

- **Registro.php:** En la **Figura 46**, este código muestra una página de registro de usuario que contiene un formulario para ingresar la información necesaria. También incorpora una función JavaScript para validar los datos y añadir al nuevo usuario mediante una solicitud AJAX.

Se incluye la librería Bootstrap para estilos y la librería jQuery para el manejo de eventos y realizar solicitudes AJAX.

Figura 46

Página de registro de usuario con un formulario

```
registro.php
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Registrar Usuario</title>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7 <link rel="stylesheet" type="text/css" href="librerias/bootstrap4/bootstrap.min.css">
8 </head>
9 <body>
10 <div class="container">
11 <h1 style="text-align: center;">Registro de usuario</h1>
12 <div class="row">
13 <div class="col-sm-4"></div>
14 <div class="col-sm-4">
15 <form id="frmRegistro" method="post" onsubmit="return agregarUsuarioNuevo()" autocomplete="off">
16 <label>Nombre</label>
17 <input type="text" name="nombre" id="nombre" class="form-control" required="">
18 <label>Apellidos</label>
19 <input type="text" name="apellidos" id="apellidos" class="form-control" required="">
20 <label>DNI</label>
21 <input type="text" name="dni" id="dni" class="form-control" required="">
22 <label>Email o Correo Electronico</label>
23 <input type="email" name="email" id="email" class="form-control" required="">
24 <label>Nombre de Usuario</label>
25 <input type="text" name="usuario" id="usuario" class="form-control" required="">
26 <label>Contraseña o Password</label>
27 <input type="password" name="password" id="password" class="form-control" required="">
28 <br>
29 <div class="row">
30 <div class="col-sm-6 text-left">
31 <button class="btn btn-primary">Registrar</button>
32 </div>
33 <div class="col-sm-6 text-right">
34 <a href="index.php" class="btn btn-success">Login</a>
35 </div>
36 </div>
37 </form>
38 </div>
39 </div>
40 </div>
41 </div>
42 <script src="librerias/jquery-3.5.1.min.js"></script>
43 <script src="librerias/sweetalert.min.js"></script>
44
```



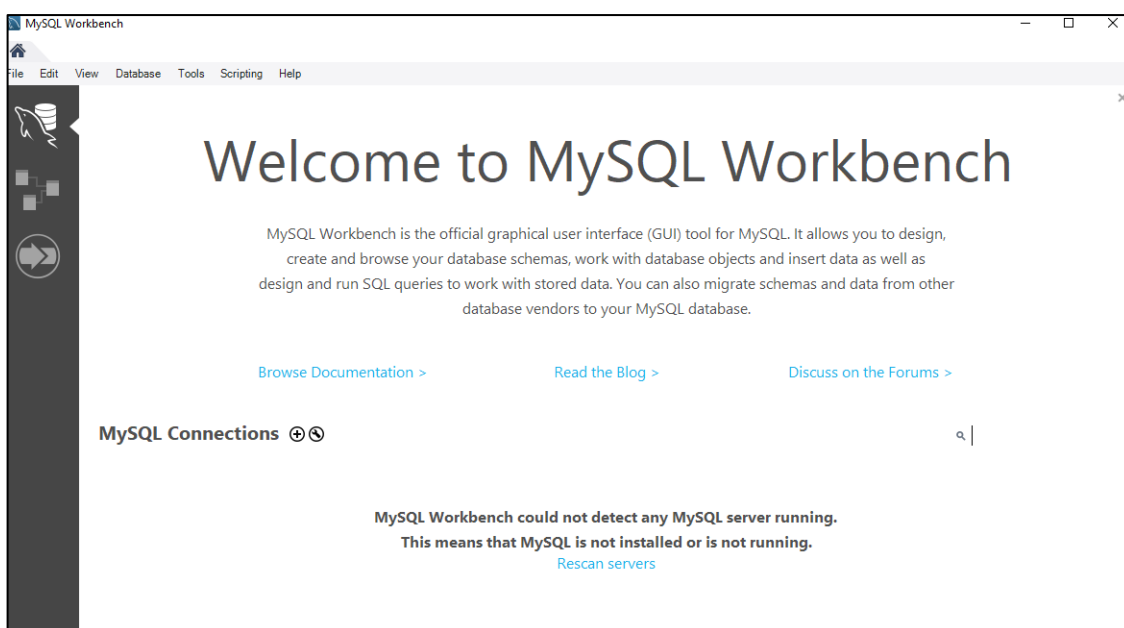
```
48     método: 'POST',
49     data: $('#frmRegistro').serialize(),
50     url: "procesos/usuario/registro/agregarUsuario.php",
51     success: function(respuesta){
52         respuesta = respuesta.trim();
53
54
55         if(respuesta == 1){
56             $('#frmRegistro')[0].reset();
57             Swal("D", "¡Agregado con éxito!", "success");
58         } else if(respuesta == 2){
59             Swal("Este usuario ya existe por favor escribe otro!");
60         } else {
61             Swal(":", "¡Falló al agregar!", "Error");
62         }
63     }
```

Dentro del <div class="container">, se encuentra el formulario de registro de usuario. El usuario debe ingresar su nombre, apellidos, DNI, correo electrónico, nombre de usuario y contraseña. Al enviar el formulario, se realiza una solicitud AJAX a través de jQuery al archivo "agregarUsuario.php" en la carpeta "procesos/usuario/registro" para agregar el nuevo usuario a la base de datos. Dependiendo de la respuesta obtenida, se muestra un mensaje de éxito si el usuario se agregó correctamente, un mensaje de error si el usuario ya existe o un mensaje de falla en caso de algún error.

- **Base de Datos:** En la **Figura 47**, se eligió MySQL Workbench como el entorno de base de datos debido a la experiencia del autor con esta herramienta. MySQL Workbench se utiliza para visualizar los diagramas de bases de datos, así como para crear las tablas y definir los atributos de cada base de datos. Luego, se establecerá la conexión entre MySQL Workbench y el entorno de desarrollo en el que se llevará a cabo la codificación del proyecto.

Figura 47

Inicio de MySQL Workbench y crear la base de datos con el nombre de local adrián

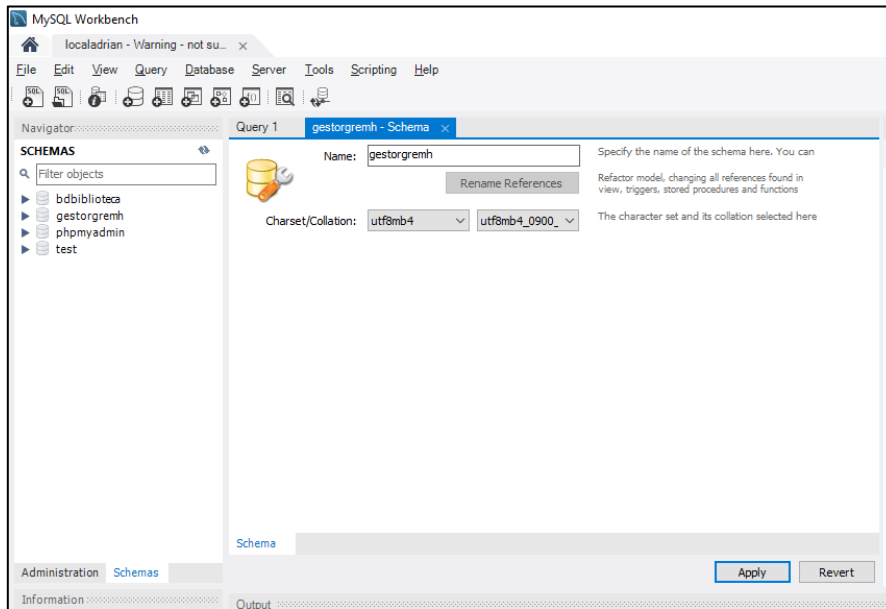




En la **Figura 48**, en este panel se procede a la creación de la base de datos denominada "gerencia". Además, se muestran las tablas que han sido creadas, que son "t_archivos", "t_categorias" y "t_usuarios".

Figura 48

Creación de la base de datos llamado "gestorgremh"



En la **Figura 49**, inicialmente se crea la tabla de usuarios con los siguientes atributos. El campo "idt_usuario" se establece como clave primaria y auto incrementable, y se agrega la función "now()" al atributo "insert" para registrar la fecha y hora por defecto en cada registro.

Figura 49

Creación de la tabla "t_usuarios"

Table: t_usuarios	
Columns:	
idt_usuario	int(11) AI PK
nombre	varchar(255)
apellidos	varchar(255)
dni	char(8)
email	varchar(245)
usuario	varchar(245)
password	text
insert	datetime

En la carpeta "clases" se crea el archivo "Conexion.php" para hacer la conexión con el MySQL, creando la clase Conectar.

En la **Figura 50**, se comienza creando la tabla "t_categorias" dentro del gestor de



MySQL, en la base de datos "gestorgremh," con los siguientes atributos: "id_categoria" (auto incremental y clave primaria), "id_usuario," "nombre," y "fechaInsert." En este último campo, se incorpora la función "now()" para que agregue la fecha automáticamente.

Figura 50

Creación de la tabla "t_categorias"

Table: t_categorias	
Columns:	
id_categoria	int(11) AI PK
id_usuario	int(11)
nombre	varchar(255)
fechaInsert	datetime

Agregar la llave foránea:

La clave externa llamada "fkCategoriaUsuario" se relaciona con la tabla "t_usuarios" a través de las columnas "id_usuario" en la tabla actual y "idt_usuario" en la tabla "t_usuarios". Esta relación permite que un usuario pueda tener muchas categorías, siguiendo un modelo uno a muchos. Esto se aplica en las Figuras 51, 52, 53 y 54.

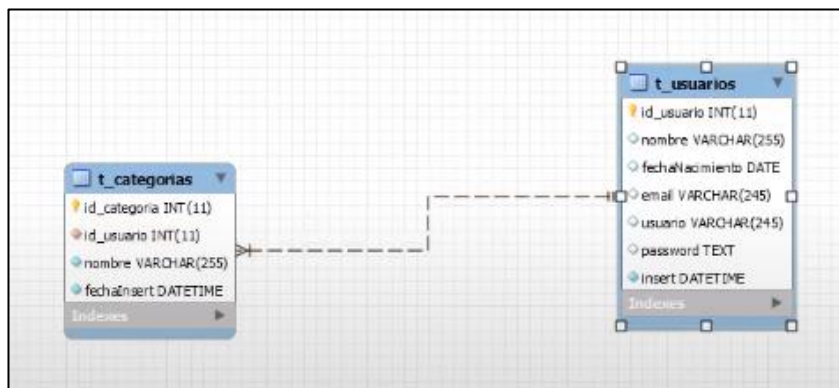
Figura 51

Llave foránea "fkCategoriaUsuario"

Foreign Key: fkCategoriaUsuario	
Definition:	
Target	t_usuarios (id_usuario → idt_usuario)
On Update	NO ACTION
On Delete	NO ACTION

Figura 52

Diagrama de clase de t_Categorias con t_usuarios



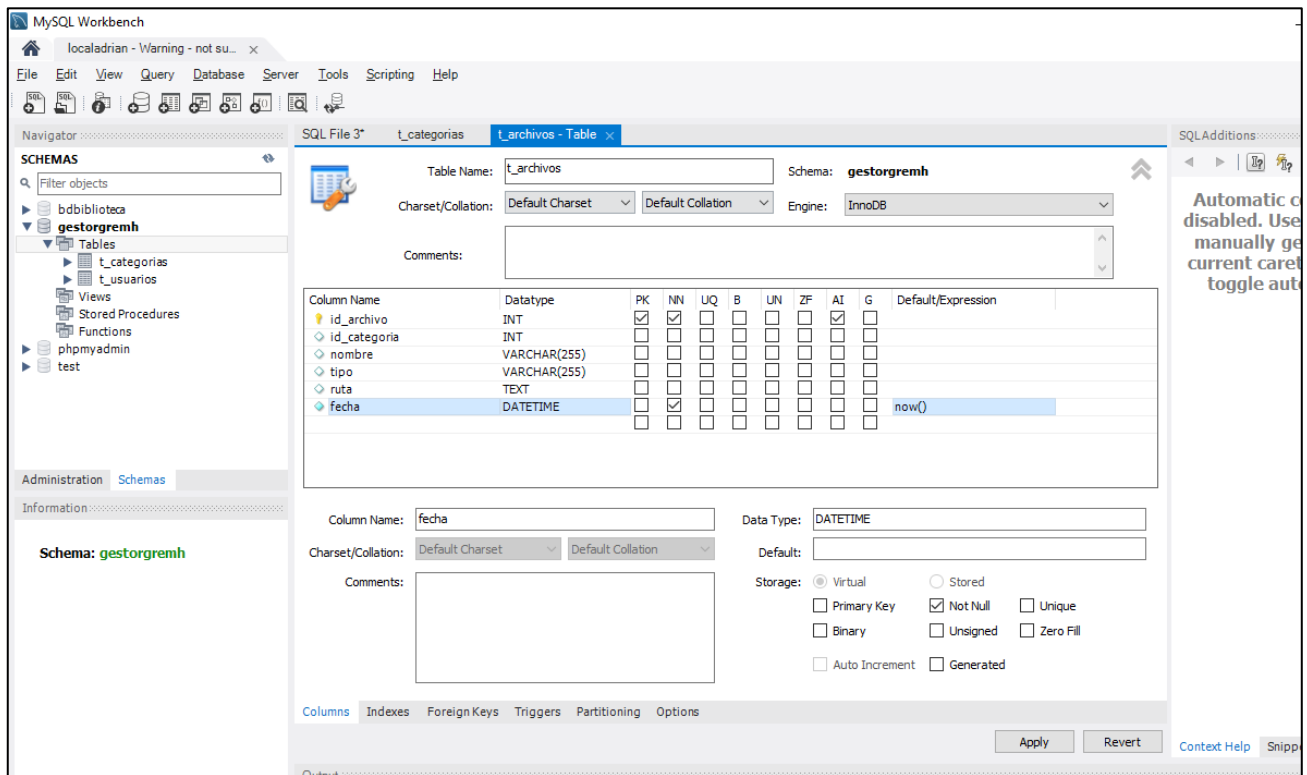


Se crea la tabla "t_Archivos" con las siguientes variables:

- id archivo (clave primaria y autoincrementar)
- id_categoria
- nombre
- tipo
- ruta
- fecha (con la función now ())

Figura 53

Creación de la tabla "t_archivos"



```

Table: t_archivos

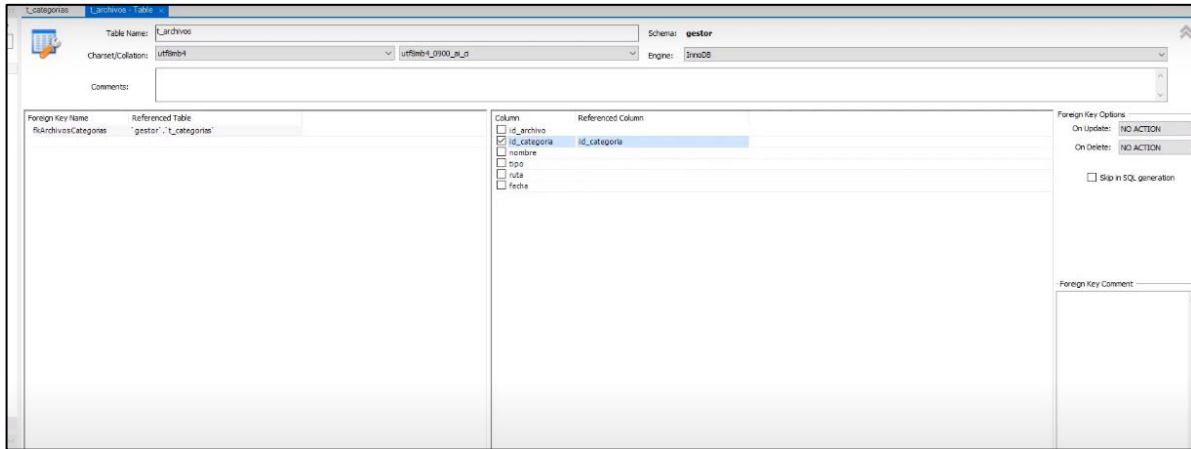
Columns:
id_archivo      int(11) AI PK
id_usuario      int(11)
id_categoria     int(11)
nombre          varchar(255)
tipo            varchar(255)
ruta            text
fecha           datetime
  
```

A continuación, se crea la llave foránea referenciando con la tabla "t_Categorias"



Figura 54

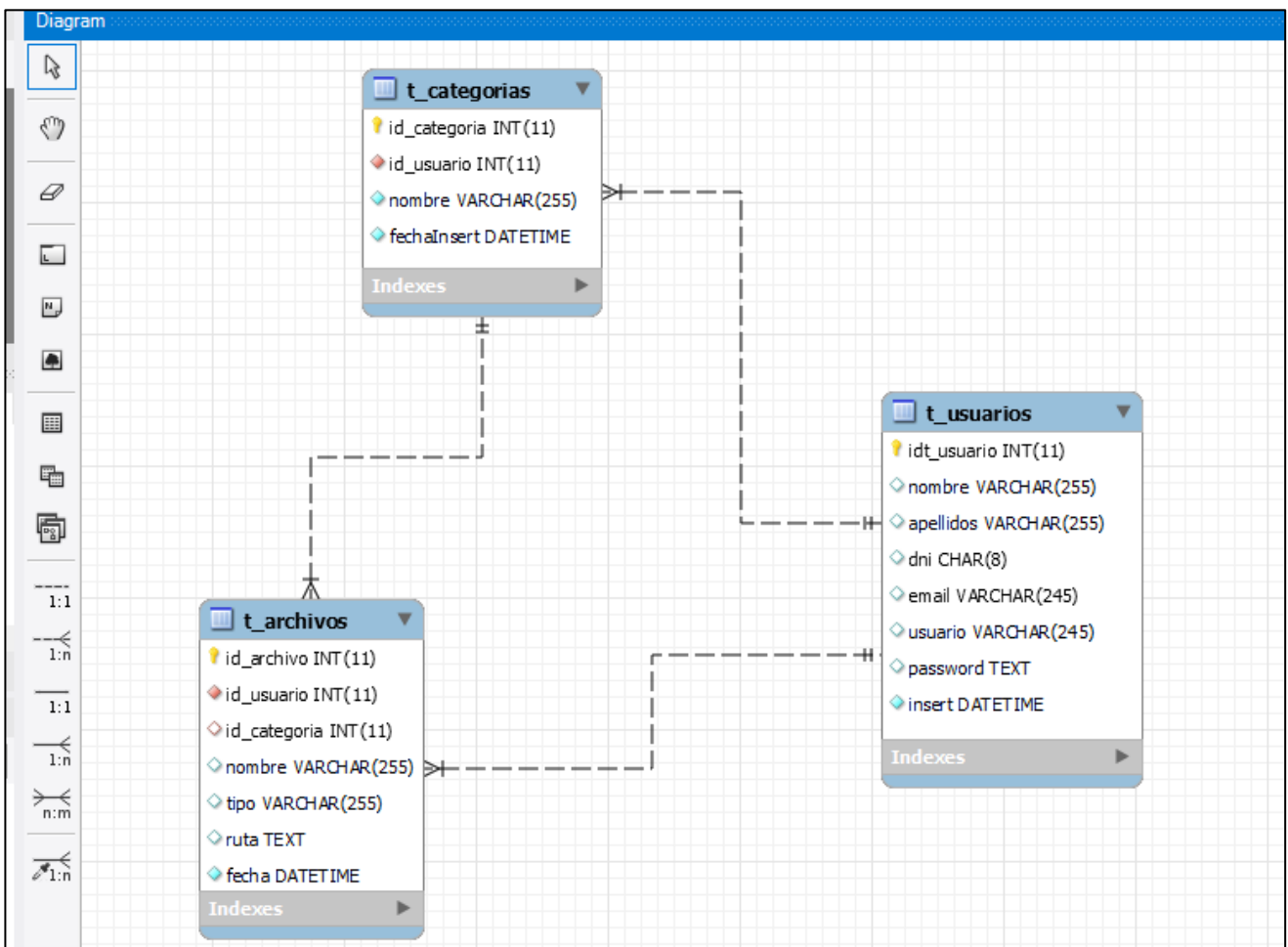
Agregar la llave foránea con la tabla “t_Categorias”



El proceso se completa con la obtención de nuestro diagrama de clases en la **Figura 55**, que presenta las tres tablas con sus relaciones y todas las variables asignadas.

Figura 55

Diagrama de clases incluyendo la relación de las tablas





2. **Pruebas unitarias:** Después de implementar cada funcionalidad, se llevan a cabo pruebas unitarias. Estas pruebas se centran en evaluar componentes o unidades de código de manera individual para verificar su correcto funcionamiento. Las pruebas unitarias desempeñan un papel crucial en la identificación y corrección temprana de errores en el código, asegurando la calidad de cada funcionalidad implementada. Las pruebas unitarias se llevan a cabo en las **Figuras 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71 y 72.**

El desarrollo del gestor de archivos tiene el siguiente orden:

Figura 56

Maquetación del Login

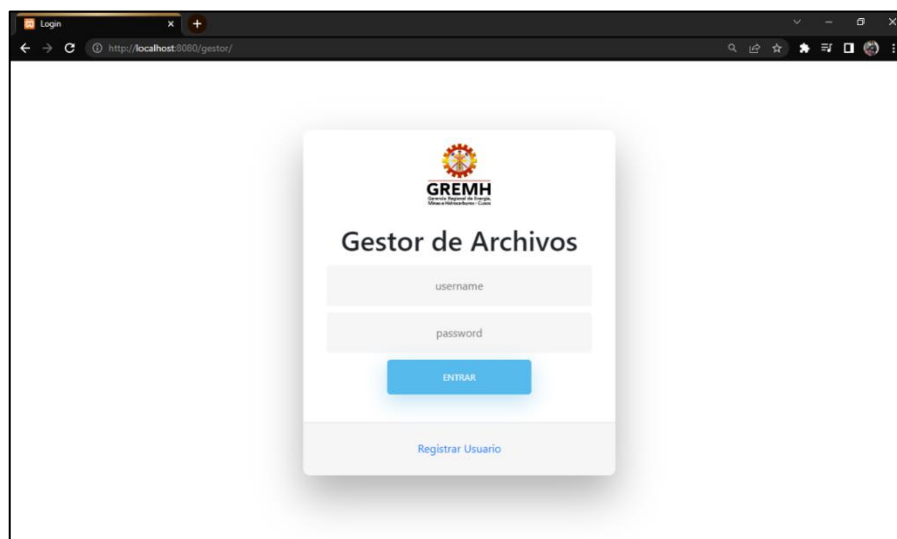


Figura 57

Crear el Registro

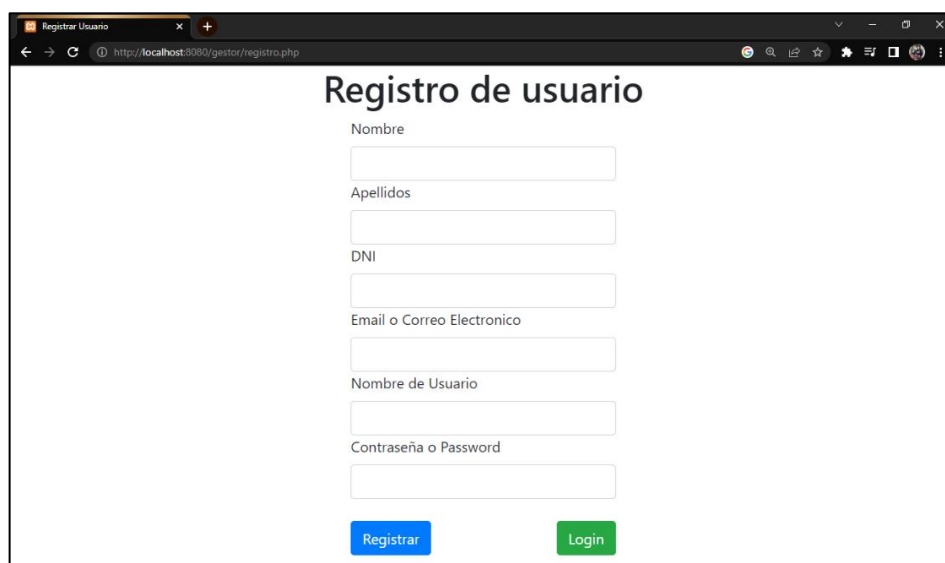




Figura 58

Desarrollo del Menú horizontal

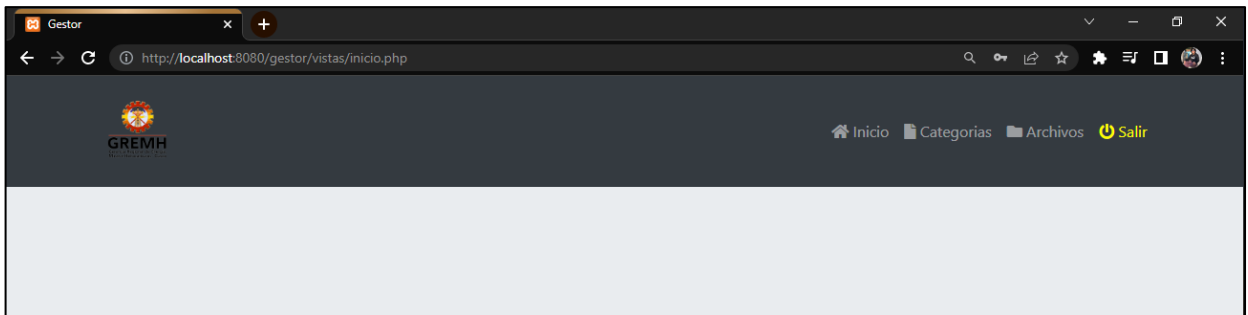


Figura 59

CRUD de archivos



Figura 60

Programación para registrar usuario

Registro de usuario

Nombre

Apellidos

DNI

Email o Correo Electronico

Nombre de Usuario

Contraseña o Password

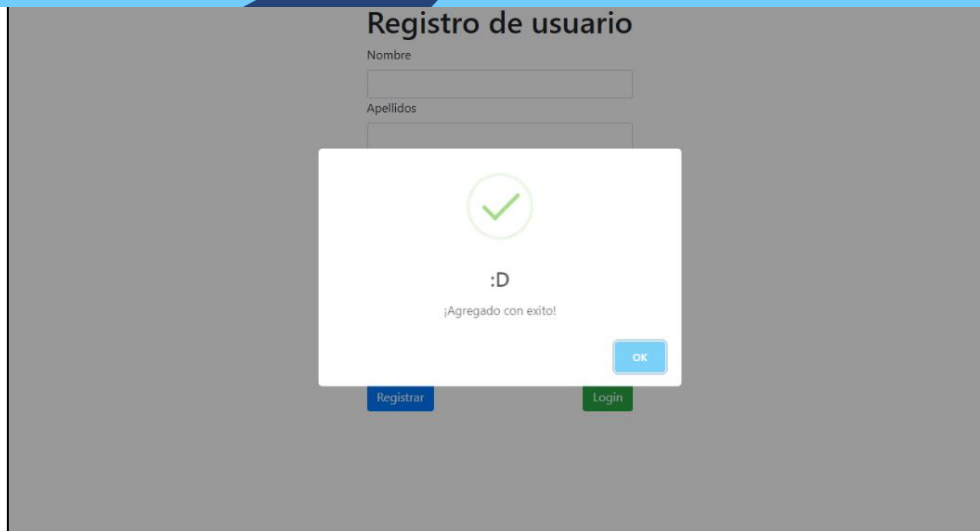


Figura 61

Verificación del nuevo usuario registrado en la base de datos "t_usuarios"

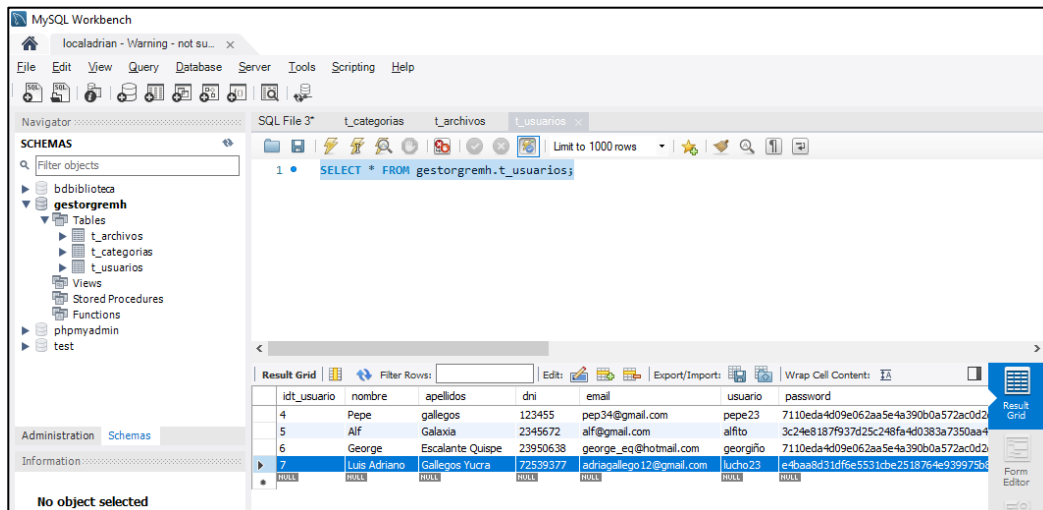


Figura 62

Programación Login de usuario

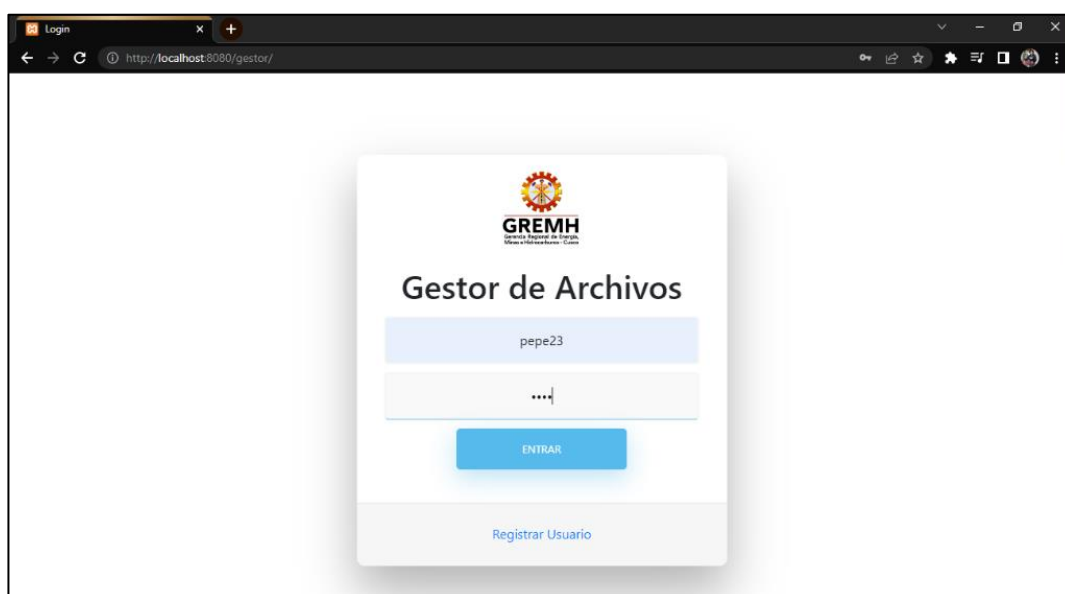




Figura 63

Implementación del Módulo Categorías

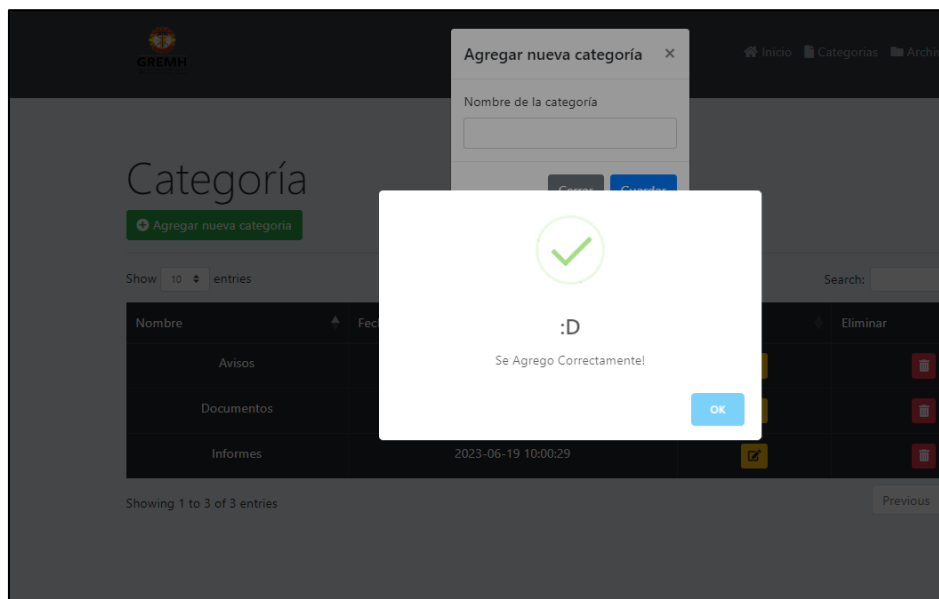
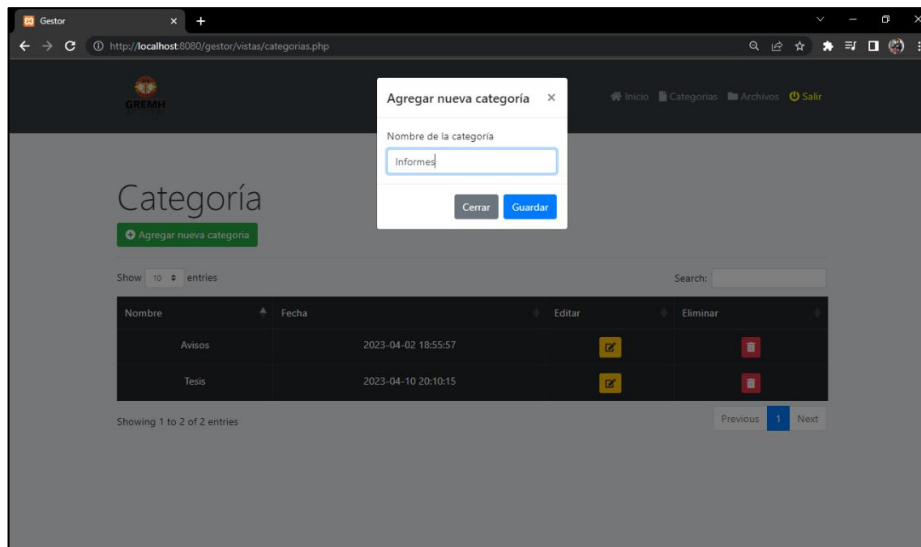




Figura 64

Verificar el registro de la nueva categoría en la base de datos

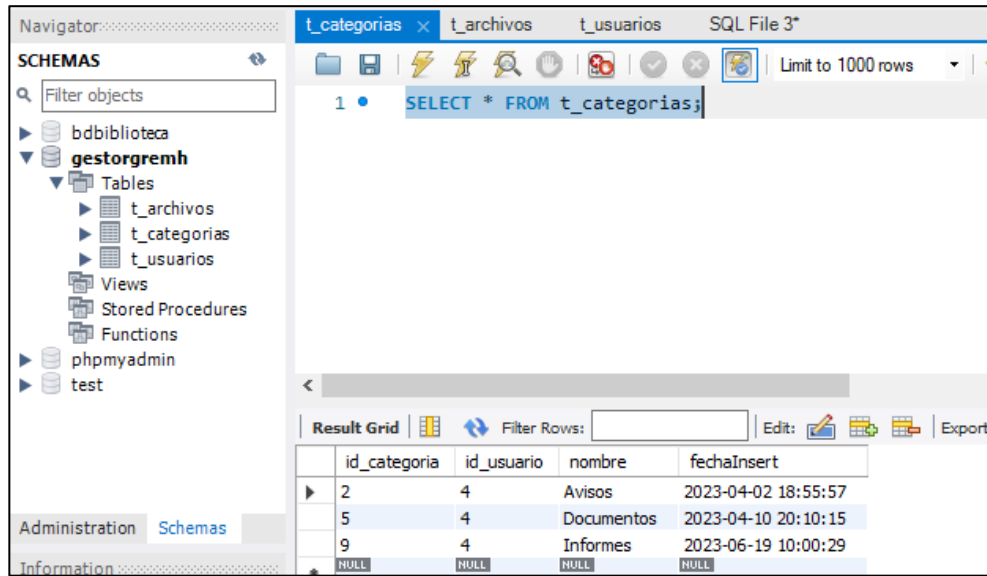


Figura 65

Módulo de Eliminar Categoría

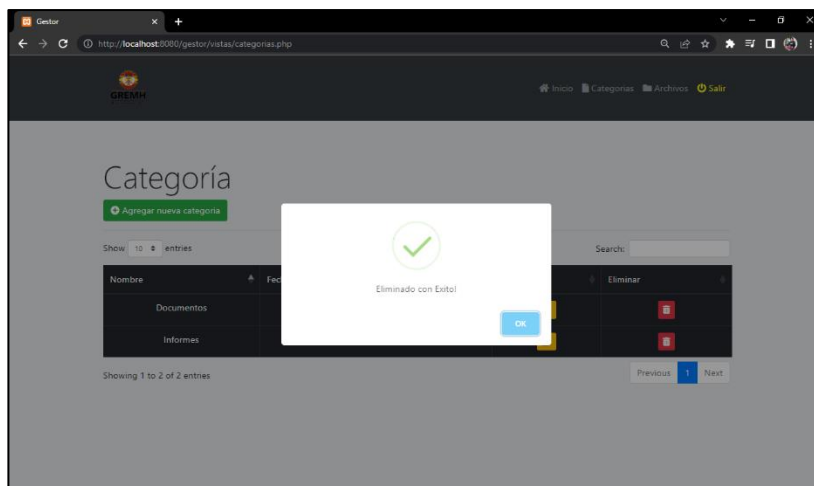
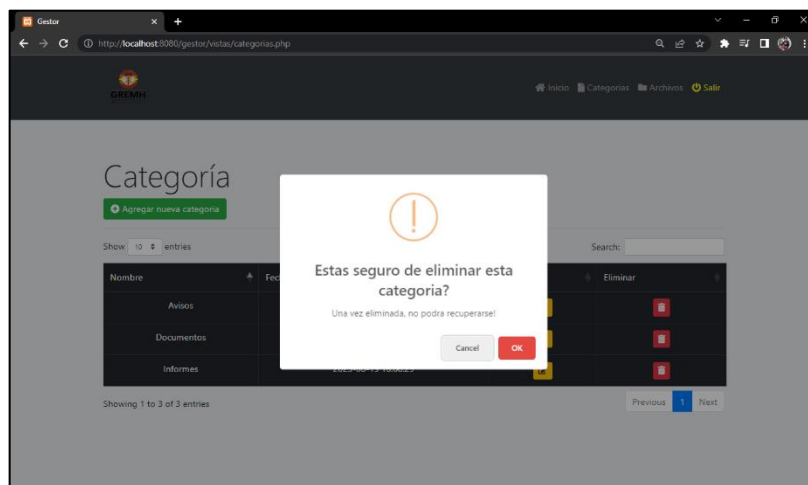




Figura 66

Módulo de Editar Categoría

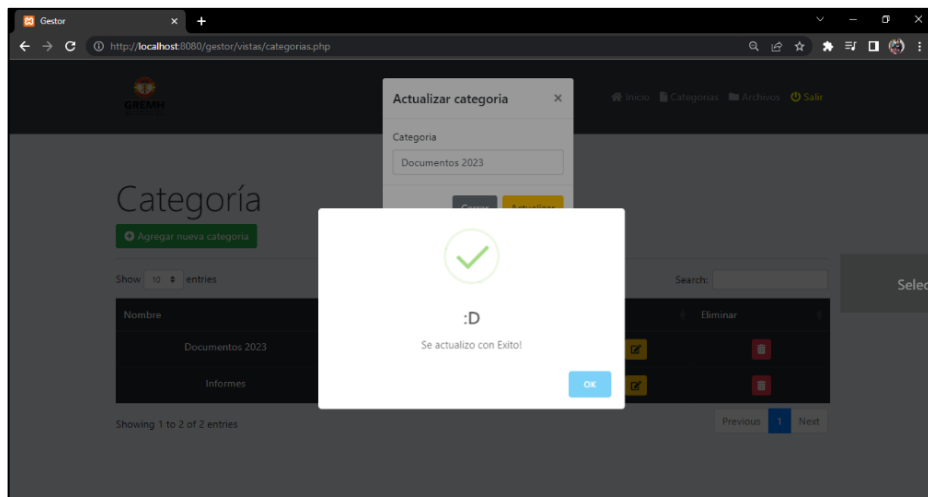
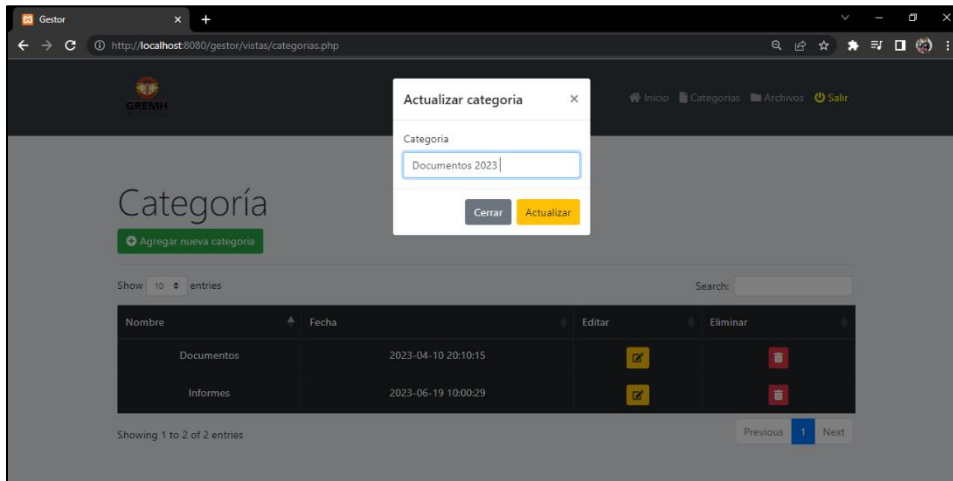


Figura 67

Guardar archivos en el módulo Archivo

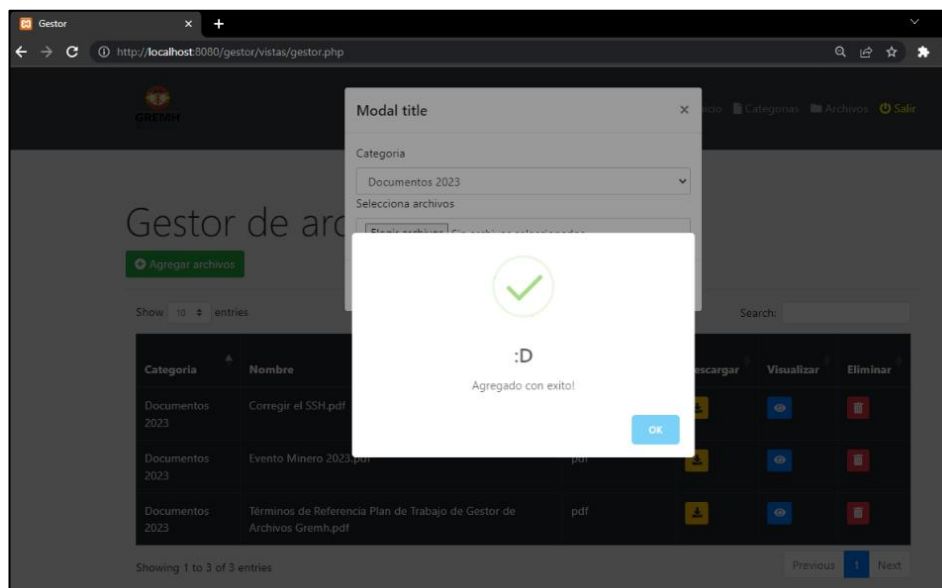




Figura 68

Verificación de los archivos en la base de datos y en la estructura del proyecto.

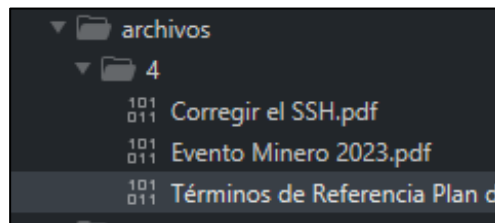
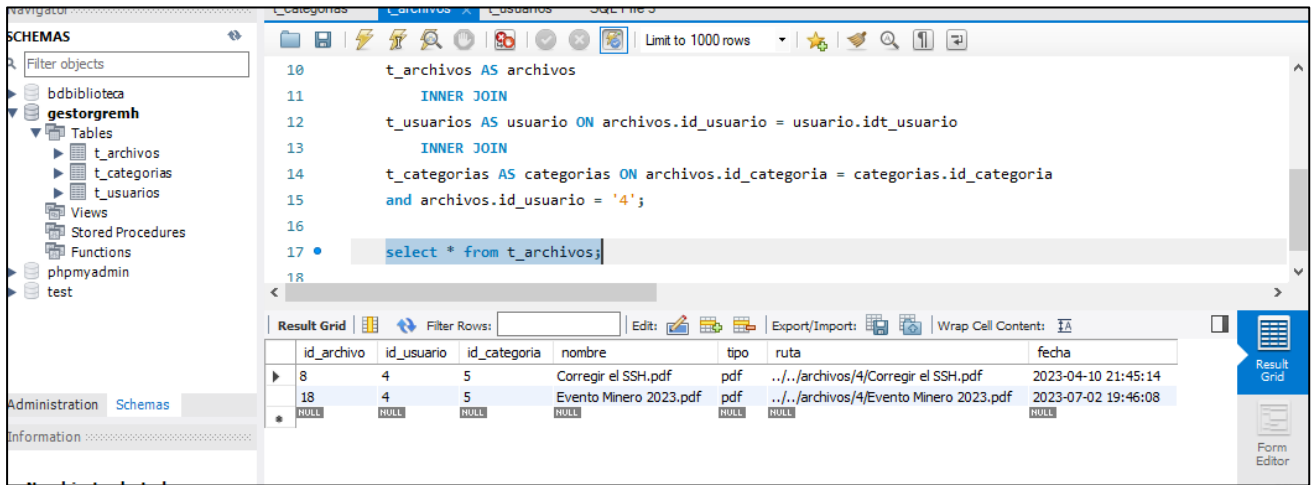


Figura 69

Funcionalidad de Descargar, Visualizar e Eliminar los archivos

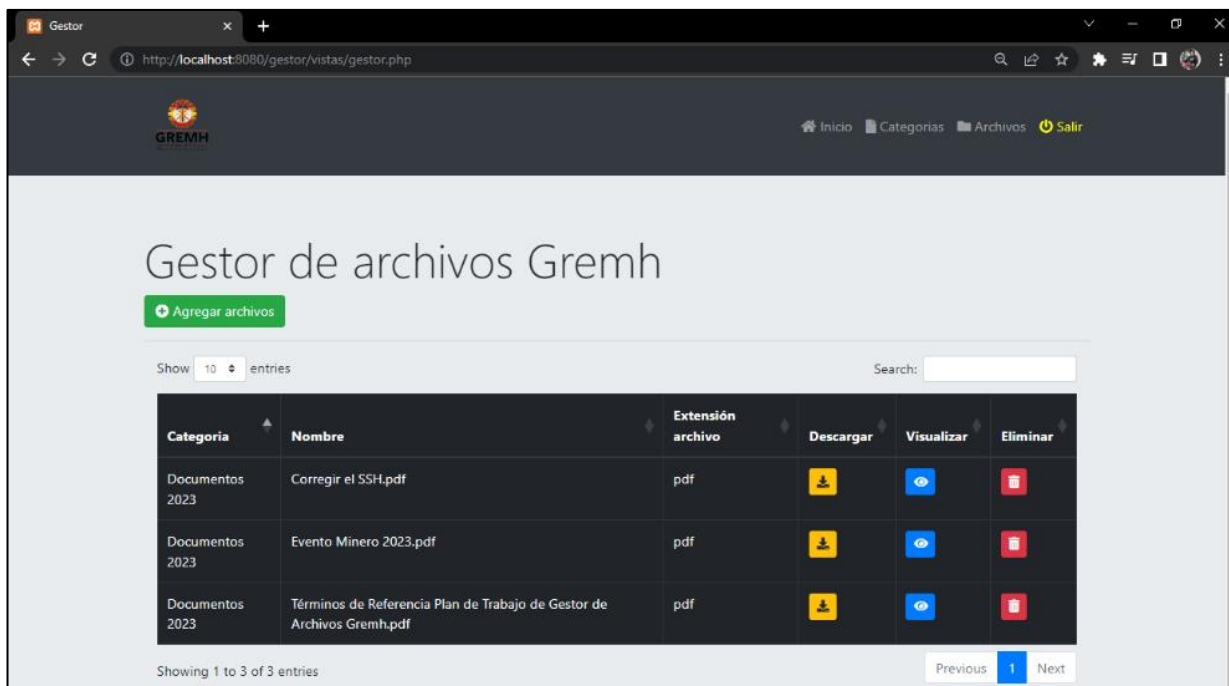




Figura 70

Descargar Archivos

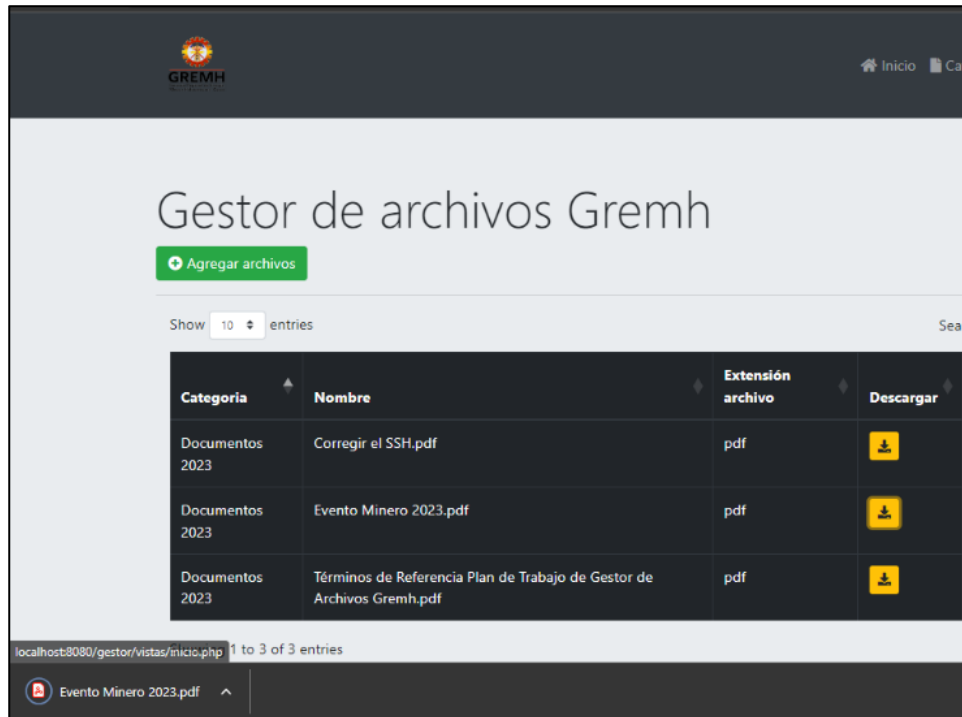


Figura 71

Visualizar Archivos

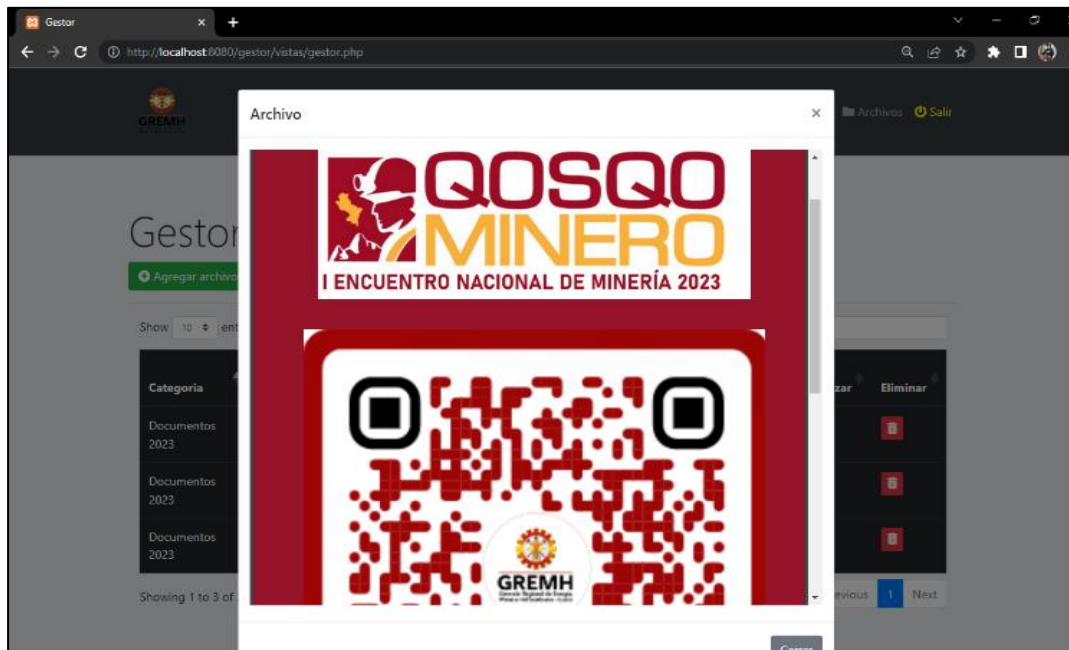
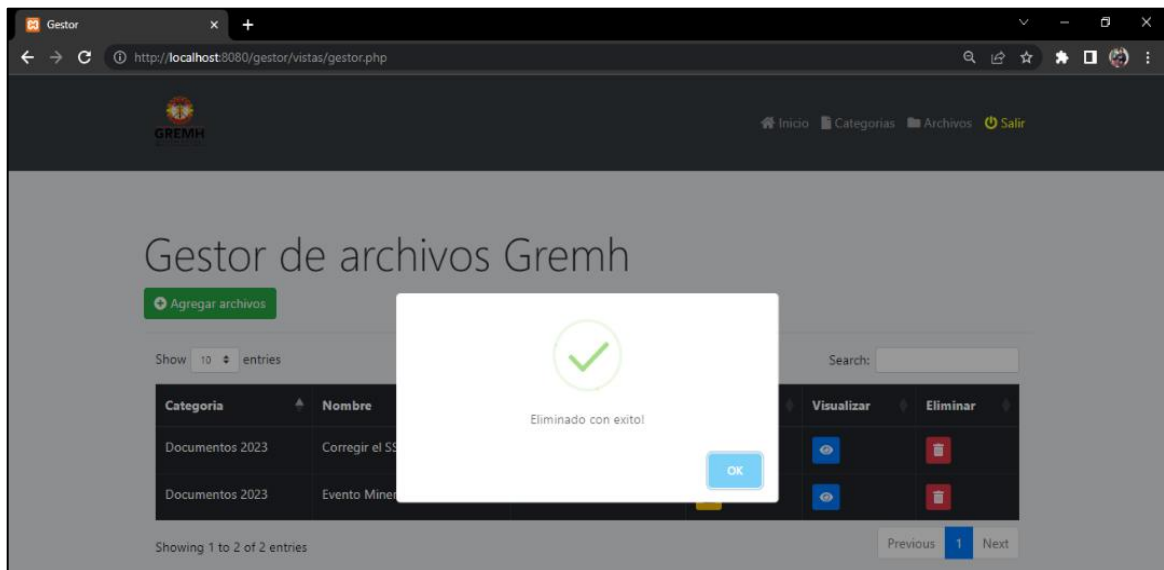


Figura 72

Eliminar Archivos



3.1.2.4 Fase 4 Pruebas

1. **Pruebas de integración:** Esta actividad también se asocia con las pruebas unitarias en la que se verifica que todas las funcionalidades del sistema de gestor de archivos se integren correctamente y funcionen en conjunto, es así que se verifica como funciona cada función del gestor de archivos así mismo la comunicación y la interacción entre los diferentes componentes del sistema para asegurarse de que trabajen de manera coherente y sin conflictos.

Digitalización, Proceso de Migración y Almacenamiento de Datos en el Sistema de Gestor de Archivos

El sistema de gestión de archivos se encuentra conectado al servidor local XAMPP y a la base de datos seleccionada denominada "gestorgremh". En la **Figura 60**, se presenta la necesidad de que el usuario se registre, y al acceder al sistema, se realiza la verificación de la existencia del nuevo usuario en la base de datos, tal como se muestra en la **Figura 61**, específicamente en la tabla "t_usuarios". Posteriormente, se dirige al usuario hacia la página de inicio de sesión, tal como se evidencia en las **Figuras 62 y 63**.

Adicionalmente, se realiza una verificación en la base de datos para corroborar la categoría creada por el usuario, seleccionando la tabla "t_categorias" en la Figura 64. Luego, se procede a cargar archivos en la categoría establecida, como se ilustra en la Figura 67. Se verifica la manera en que se registró el nombre del archivo en



la base de datos mediante la selección de la tabla "t_archivos".

Añadiendo a esto, es importante destacar que el almacenamiento de la información se realizará tanto en el servidor local XAMPP como en la base de datos MySQL Workbench. Esta decisión asegura una gestión eficiente de los datos, permitiendo su acceso y recuperación de manera rápida y segura.

Además, el proceso implica la digitalización de documentos en el escáner de la gerencia por parte del personal interno del departamento de informática, en colaboración con quien suscribe. Este paso asegura que los documentos físicos se conviertan en archivos digitales antes de ser cargados en el sistema.

En resumen, el proceso completo involucra:

1. Registro de un nuevo usuario.
2. Ingreso al sistema y verificación en la base de datos.
3. Creación y verificación de la categoría creada por el usuario.
4. Digitalización de documentos en el escáner.
5. Subida de archivos a la categoría creada y verificación en la base de datos.
6. Almacenamiento de la información en el servidor local y en la base de datos.
7. Verificación de los archivos cargados por el usuario.

Capacidad de Almacenamiento Actual:

El servidor ha sido adaptado para facilitar la carga de archivos de mayor tamaño, mediante el ajuste del límite de tamaño de carga (`post_max_size = 1024M`). En esta configuración específica, se ha establecido la capacidad para manejar archivos PDF de hasta 1 GB. Esta modificación se detalla en el **anexo 2**, específicamente en la **Tabla 13**, que proporciona una visión general del presupuesto para el proyecto de investigación y desarrollo del sistema de gestor de archivos.

- 2. Pruebas de aceptación:** Se verifica las pruebas de aceptación por cada historia de usuario que se definió en las fases anteriores. Estas pruebas se centran en verificar que cada funcionalidad cumpla con los criterios de aceptación establecidos y satisfaga las necesidades del cliente, esto se puede verificar en las siguientes tablas: **Tabla 1,2,3 y 4**.
- 3. Corrección de errores:** Durante las pruebas, es probable que se encuentren errores o fallos en el sistema. En esta actividad, se analizan y se corrigen los errores



identificados. La corrección de errores es un proceso iterativo que implica la identificación, solución y verificación de los problemas encontrados para asegurar que el sistema esté libre de errores antes de su implementación final.

3.1.2.5 Fase 5 Integración continua:

- 1. Validación de la satisfacción del cliente:** Se realiza una revisión final del software por parte de la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco en el área de archivos para validar que el software cumple con las necesidades del negocio.
- 2. Preparación para la entrega:** Se prepara el software para su entrega al cliente, incluyendo la documentación necesaria, manuales de usuario, instaladores, entre otros.
- 3. Entrega del software:** Se entrega el software a la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco en el área de archivos, junto con el manual necesario para su uso.

Link:

https://drive.google.com/file/d/1Sv17ZIITWoHOs6CGCQ1NkT5pqjk4wq94/view?usp=drive_link

En la **Figura 73** se muestra la ubicación física del área de archivos de la Gremh, mientras que en la **Figura 74** se presenta el manual de uso del sistema gestor de archivos. Estos elementos son esenciales para proporcionar a los usuarios acceso a los recursos físicos y a las instrucciones necesarias para utilizar el sistema de gestor de archivos.

Figura 73

Local del área de archivos de la oficina de administración





Figura 74

Manual de uso del sistema de Gestor de Archivos Gremh (<https://goo.su/UCIMbOX>)



Las **Figuras 75, 76 y 77** muestran entrevistas relacionadas con las áreas que componen la Gremh y sesiones de capacitación sobre el uso del sistema de gestor de archivos. Estas etapas son esenciales para comprender las necesidades de los usuarios y garantizar una implementación exitosa del sistema.

Figura 75

Área de Minería y Concesión minera



Figura 76

Área de cobertura energética y sistema de ventanilla única para formalización minera



Figura 77

Área de Proyectos





CAPÍTULO 4 – Resultados

4.1 Comprobación de la prospectiva

Se consideran dos casos:

- **N°1:** Estudio del área de archivos sin el sistema de gestor de archivos.
- **N°2:** Estudio del área de archivos con el sistema de gestor de archivos

Evaluación: Dentro de la Gremh, se trabajan en diversas áreas. Una vez que varios proyectos concluyen, se procede a revisarlos y almacenarlos en el archivo correspondiente. Estos archivos contienen información sumamente relevante para la gerencia, incluyendo el título del proyecto, la respectiva área a la que pertenece, el año de gestión y el mes en el que se llevó a cabo, para luego almacenar en el área de archivos.

Prospectiva:

- **Futuro tendencial para área de archivos con el sistema de gestor de archivos (N°1):**
 - **Desorganización y Pérdida de Información:** Sin un sistema de gestor de archivos, existe un alto riesgo de desorganización y pérdida de documentos importantes. La falta de una estructura organizativa podría dificultar la búsqueda y recuperación de información crítica en el futuro.
 - **Ineficiencias en la Gestión:** La falta de un sistema automatizado podría llevar a ineficiencias en la gestión de documentos y proyectos. Los procesos manuales para mantener y actualizar los archivos podrían ser más lentos y propensos a errores.
 - **Mayor Uso de Espacio Físico:** Sin una solución de gestor de archivos electrónicos, se podría requerir un mayor espacio físico para almacenar la documentación en papel, lo que podría resultar en costos adicionales asociados al espacio y la administración.
- **Futuro deseable para área de archivos con el sistema de gestor de archivos (N°2):**
 - **Eficiencia y Productividad Mejoradas:** El sistema de gestor de archivos optimizaría los procesos de búsqueda, recuperación y organización de documentos, lo que resultaría en una mayor eficiencia y productividad. Esto permitiría una utilización más efectiva del tiempo y los recursos.



- **Acceso Rápido y Seguro a la Información:** Los usuarios tendrían acceso inmediato y seguro a la información relevante en cualquier momento. Esto agilizaría la toma de decisiones y la colaboración entre equipos y áreas.
- **Reducción de Costos:** La implementación de un sistema de gestor de archivos electrónico reduciría la dependencia de documentos en papel y, por lo tanto, disminuiría los costos asociados con el almacenamiento físico, la impresión y la administración manual.

Estrategia:

- Desarrollo e integración del módulo de registro en el sistema.
- Incorporación del módulo de inicio de sesión al sistema.
- Despliegue del módulo de creación de categorías.
- Implementación del módulo de edición de tipos de categoría.
- Incorporación del módulo para guardar archivos dentro de categorías.
- Desarrollo del módulo de edición de archivos en categorías.
- Integración del módulo de visualización de archivos en categorías.
- Despliegue del módulo de descarga de archivos desde categorías.
- Implementación del módulo para eliminar archivos de categorías.
- Incorporación del módulo de búsqueda de archivos en categorías.

Desarrollo:

El proceso de investigación que llevó al desarrollo del sistema de gestor de archivos comenzó en 2023. A lo largo de este período, se recopiló información, se realizaron entrevistas en el **Anexo 1** para detalles adicionales y se tomaron fotografías, en las **Figuras 75, 76 y 77**, en las diversas áreas que integran la Gremh. El enfoque primordial de esta investigación fue implementar un sistema diseñado para automatizar la gestión de información en el ámbito de archivos.



Aportes de los Antecedentes Nacionales e Internacionales en el Trabajo de Investigación.

Tabla 11

Aportes de los Antecedentes

Aportes de los Antecedentes Nacionales	Los antecedentes nacionales validaron la relevancia de los problemas de gestión documental en mi contexto, ofrecieron ejemplos de problemáticas comunes, resaltaron la adaptación a normativas locales y brindaron valiosas lecciones sobre el cumplimiento legal, enriqueciendo así mi investigación y mejorando la calidad de mi proyecto.	<p>Aportes de los Antecedentes Nacionales:</p> <ul style="list-style-type: none"> • Relevancia Local: Los antecedentes nacionales demostraron que los problemas de gestión documental son relevantes a nivel nacional, lo que validó la importancia de mi investigación en mi contexto local. • Identificación de Problemáticas Comunes: Estos antecedentes permitieron identificar problemáticas comunes en la gestión documental a nivel nacional, como ineficiencias en el manejo de documentos y la necesidad de mejorar la eficiencia y la organización. • Elección de Tecnologías y Métodos: Conocer las tecnologías y métodos utilizados en investigaciones nacionales previas influyó en mi elección de enfoques y herramientas para el desarrollo de mi propio sistema de gestión documental.
Aportes de los Antecedentes Internacionales	Los antecedentes internacionales ampliaron mi perspectiva global sobre los problemas de gestión documental, inspirándome con soluciones innovadoras en	<p>Aportes de los Antecedentes Internacionales:</p> <ul style="list-style-type: none"> • Ampliación de la Perspectiva: Los antecedentes internacionales ampliaron mi comprensión al mostrar cómo se han abordado los problemas de gestión documental en diversos contextos y países. • Inspiración para Soluciones Innovadoras: Estos antecedentes proporcionaron ideas innovadoras y soluciones exitosas que



diversos contextos. También destacaron la importancia del cumplimiento de normativas legales y proporcionaron valiosas lecciones, enriqueciendo mi investigación y brindándome una visión global de la gestión documental.

inspiraron el diseño y desarrollo de mi propio sistema de gestión documental.

- **Cumplimiento Legal Global:** Algunos antecedentes se centraron en el cumplimiento de normativas legales a nivel internacional, lo que me ayudó a comprender la importancia de garantizar la conformidad legal en mi propio proyecto.
- **Tecnologías y Métodos Globales:** Conocer las tecnologías y métodos utilizados en investigaciones internacionales previas influyó en mi elección de enfoques y herramientas para el desarrollo de mi sistema de gestión documental a nivel global.

4.2 Cumplimiento de objetivos

Tabla 12

Cumplimiento de objetivos

Objetivos	Descripción
Objetivo General: Implementar un sistema de gestor de archivos basado en la tecnología Bootstrap en el área de archivos de la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco	Para cumplir con el objetivo general, se ha implementado el sistema de gestor de archivos en el área de archivos con la colaboración del personal del departamento de informática. Además, se ha desarrollado una guía de capacitación sobre el manejo y uso del sistema, lo que ha permitido que el sistema esté operativo para la gestión eficiente de documentos. Paralelamente, se ha proporcionado un manual de uso del sistema de gestor de archivos Gremh.
Primer Objetivo: Identificar los requisitos del usuario a través de entrevistas con los usuarios y el equipo de la	Para cumplir con el Primer Objetivo, se llevaron a cabo entrevistas con el área de Administración, que es el departamento responsable de la gestión. Además, se tomaron fotografías del área de archivos. Esta recopilación de información brindó un mayor conocimiento sobre las



Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco. funciones de cada área que opera en la gerencia, así como los procedimientos utilizados para almacenar los documentos de proyectos en el área de archivos. Estos datos fueron fundamentales para elaborar las historias de usuario necesarias para el proyecto.

Segundo Objetivo: Como segundo objetivo, se realizó la maquetación del proyecto y se organizó la construcción del diagrama de casos de uso, así como la creación de las historias de usuario y el diagrama de la arquitectura del proyecto. Luego, se seleccionaron las tecnologías a utilizar y se diseñaron las interfaces de usuario. A continuación, se procedió a la codificación de manera organizada, implementando el proyecto por etapas y realizando las pruebas correspondientes de funcionalidad, así como modificaciones y correcciones en el código.

Tercer Objetivo: Como último objetivo, se evaluó que todas las interfaces funcionaran correctamente y se procedió a desplegar el sistema en la gerencia, adjuntando un manual de uso del sistema para ponerlo en funcionamiento.

Realizar el despliegue del sistema de gestor de archivos utilizando la tecnología Bootstrap en la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco.

4.3 Contribuciones (impacto)

La implementación del sistema de gestor de archivos en la Gerencia Regional de Energía, Minas e Hidrocarburos en Cusco aportará diversas contribuciones significativas al área de archivos, mejorando la eficiencia de las operaciones. Algunas de estas contribuciones incluyen:

- **Organización y Acceso Eficiente:** El sistema permitirá una organización estructurada de los archivos, facilitando la categorización y el almacenamiento ordenado de la información. Esto conducirá a un acceso rápido y eficiente a los documentos necesarios, reduciendo el tiempo necesario para buscar información y mejorando la productividad.



- **Gestión de Información Completa:** El sistema de gestor de archivos proporcionará una visión integral de la información en el área. Facilitará el seguimiento de la documentación relacionada con proyectos, decisiones y actividades, permitiendo un entendimiento completo de los procesos y eventos pasados.
- **Ahorro de Espacio y Recursos:** Al reducir la dependencia de documentos en papel, el sistema de gestor de archivos contribuirá al ahorro de espacio físico y recursos asociados con el almacenamiento y manejo de documentos en papel.
- **Seguridad y Confidencialidad:** El sistema ofrecerá opciones de seguridad avanzadas para proteger los documentos sensibles y confidenciales. Los accesos y permisos controlados garantizarán que solo las personas autorizadas puedan acceder a la información pertinente.



Glosario

- **Bootstrap:** Bootstrap es un framework de diseño web de código abierto desarrollado por Twitter. Proporciona una colección de herramientas y estilos CSS predefinidos que facilitan la creación de sitios web receptivos y estéticamente atractivos.
- **Clases de CSS:** Bootstrap utiliza clases de CSS predefinidas para aplicar estilos y diseños a los elementos HTML. Por ejemplo, las clases "btn" y "btn-primary" se utilizan para crear botones estilizados con el color primario de Bootstrap.
- **Contenedor (Container):** Es un elemento HTML utilizado para envolver y limitar el ancho del contenido en una página web. Bootstrap ofrece diferentes tipos de contenedores, como "container" y "container-fluid".
- **Diagrama UML:** Representación gráfica estandarizada utilizada en ingeniería de software y diseño de sistemas para visualizar, especificar, construir y documentar diferentes aspectos de un sistema.
- **Grid (Cuadrícula):** La cuadrícula de Bootstrap es un sistema de diseño basado en filas y columnas que ayuda a organizar el contenido de manera estructurada y flexible en una página web.
- **HTML:** (HyperText Markup Language) es el lenguaje de marcado utilizado para crear páginas web. PHP se integra con HTML para generar contenido dinámico dentro de las páginas web.
- **MySQL:** Es un sistema de gestión de bases de datos relacional muy popular. PHP se integra fácilmente con MySQL para almacenar y recuperar datos de bases de datos.
- **OOP (Programación Orientada a Objetos):** PHP admite la programación orientada a objetos, lo que permite organizar el código en clases y objetos para facilitar la reutilización y el mantenimiento.
- **PHP:** Hypertext Preprocessor. Es un lenguaje de programación del lado del servidor utilizado para crear aplicaciones web dinámicas e interactivas.
- **Receptivo (Responsive):** Un diseño web receptivo (responsive) es aquel que se adapta y se ve bien en diferentes dispositivos y tamaños de pantalla, como computadoras de escritorio, tabletas y teléfonos móviles.
- **Script:** Un programa o código escrito en PHP que se ejecuta en el servidor web para producir contenido dinámico.
- **Servidor web:** Un software que se ejecuta en un servidor y se encarga de procesar las solicitudes HTTP, como Apache.



Conclusiones

- La implementación de un sistema de gestor de archivos basado en la tecnología Bootstrap en la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco, específicamente en el área de archivos, representa un paso importante para mejorar la organización y eficiencia en la gestión de documentos en esta entidad. Este sistema facilitará la administración, acceso y seguimiento de los archivos, lo que a su vez ayudará a optimizar los procesos internos.
- La identificación de los requisitos a través de entrevistas con los usuarios y el equipo de la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco, el sistema de gestor de archivos se ha adaptado a las necesidades específicas del usuario. La participación activa de los usuarios en el proceso de desarrollo ha llevado a un producto final que cumple con sus expectativas y mejora la eficiencia en el gestor de archivos.
- El enfoque iterativo e incremental utilizado en el diseño y desarrollo del sistema de gestor de archivos ha demostrado ser efectivo para crear un producto robusto y funcional. Este enfoque ha permitido realizar ajustes y mejoras a lo largo del proceso de desarrollo, asegurando que el sistema final sea estable y de alta calidad.
- La adopción de tecnologías y enfoques modernos, como Bootstrap y el desarrollo iterativo, ha contribuido al éxito del despliegue, permitiendo un sistema más ágil y adaptado a las necesidades cambiantes del entorno de la Gerencia Regional de Energía, Minas e Hidrocarburos – Cusco, asegurando que el sistema esté disponible y accesible para su uso diario.



Recomendaciones

- Se recomienda a los usuarios que utilicen el manual de usuario como su principal fuente de referencia para aclarar cualquier duda relacionada con el funcionamiento del sistema de gestor de archivos. El manual proporciona información detallada sobre los pasos que deben seguirse para garantizar un uso correcto y eficiente del sistema. La consulta de este recurso no solo facilitará la comprensión de las funcionalidades disponibles, sino que también contribuirá a optimizar la experiencia general de gestor de archivos.
- Se recomienda que se lleven a cabo capacitaciones periódicas. A pesar de que el sistema esté en funcionamiento, es crucial continuar brindando capacitación y soporte técnico a los usuarios para maximizar su aprovechamiento. Algunos usuarios podrían necesitar asistencia adicional para sacar el máximo provecho de las funcionalidades del sistema, por lo que se deben programar sesiones de capacitación regulares.
- Se recomienda el establecimiento de un proceso de mantenimiento y actualización periódica. Esto involucra asignar recursos y un equipo responsable de supervisar el funcionamiento del sistema, detectar posibles problemas y llevar a cabo mejoras continuas. Dado que la tecnología y las necesidades del usuario evolucionan con el tiempo, mantener el sistema actualizado asegurará que siga siendo relevante y eficiente en el futuro. Además, el mantenimiento periódico permitirá abordar cualquier vulnerabilidad de seguridad que pueda surgir y garantizar la integridad y confidencialidad de los archivos gestionados.
- Se recomienda establecer un proceso de monitoreo continuo del rendimiento del sistema de gestor de archivos. Este proceso implica la supervisión constante del uso del sistema, el tiempo de respuesta, la carga del servidor y otros indicadores clave para identificar posibles cuellos de botella o problemas de rendimiento. Con base en los datos recopilados, se deben llevar a cabo ajustes y optimizaciones con el fin de asegurar que el sistema funcione de manera eficiente y sin interrupciones.



Referencias

- Aquino Espinoza, Gerson. (2019). Desarrollo e Implementación del Sistema de Gestión de Documentos ScanViewer Para la Empresa Global Factoring S.A. [Trabajo de Investigación, Universidad Tecnológica del Perú] <https://goo.su/svZqe5T>
- Bastidas, J. (2016) realizo el proyecto de investigación Desarrollo e Implementación del Sistema de Tramite Documentario en la Municipalidad Provincial de Huancayo para la atención de expedientes. Realizado en Perú, Recuperado de <https://goo.su/hekgt>
- Catalina Lacunza, Ana. (2021). Implementación de un sistema de gestión documental electrónico en la UNLP. [Trabajo de Investigación, Universidad Nacional de la Plata] <https://goo.su/nL90>
- Centeno, E. y Cordones, S. (2016) realizo la investigación “Implementación de un Sistema de Gestión Documental Administrativa aplicando la Tecnología Php Bajo el Framework Codeigniter, Html5, Css3 y Mysql para la Unidad Académica de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi en el Periodo 2015 – 2016”. Realizado en Ecuador. Recuperado de <https://goo.su/oB1RnS>
- Coello, D., & Izquierdo, J. L. (2008). Tutorial de Prácticas del Gestor de Bases de Datos MySQL. Recuperado de <https://goo.su/y2fzI>
- Cordona Valdes, J.A, Osorio Lopez, O.L. (2019). Diseño e Implementación de un Sistema de Gestión Documental para el cumplimiento de la ley 1712 de 2014. [Trabajo de Investigación, Universidad Tecnológica de Pereira] <https://goo.su/XL0oQ>
- Díaz Rodríguez, María del Rosario. (2009). Los archivos y la Archivística a través de la historia. <https://goo.su/sq5v>
- Eguiluz, J. (2008). Introducción a JavaScript. Recuperado de: <https://goo.su/gT4N9O>
- Fenoglio, Norma Catalina. (2013). Teoría de la macro evaluación de los documentos de archivo. <https://goo.su/UBULG>
- Gallejo, J. (2003). Desarrollo Web con Php y MySQL. Editorial ANAYA MULTIMEDIA (GRUPO ANAYA) S. A. Recuperado de <https://goo.su/Umwd2Wx>
- Gerencia Regional de Energía, Minas e Hidrocarburos Cusco | Facebook. (s. f.). Recuperado 3 de mayo de 2023, de <https://www.facebook.com/GremhGoreCusco/>
- Gutierrez Argomedeo, S. A. (2019). Automatización de la gestión archivística para



- optimizar los pedidos de documentos del área de archivo de la Ugel 03 [Trabajo de Investigación, Universidad Científica del Sur] <https://goo.su/n1TB>
- Izquierdo, I (2015) realizo el proyecto de investigación “Sistema cero papeles para la gestión documental del grupo de intervención y rescate Gir”. Realizado en Ecuador. Recuperado de: <https://goo.su/U9mF>
- Jakobus, B., & Marah, J. (2017). Domina la última versión de Bootstrap 4 para crear aplicaciones web responsivas altamente personalizadas. <https://goo.su/goWiY>
- Lujan, J. (2002). Programación de Aplicaciones web: historia, principios básicos y clientes web. Editorial Club Universitario. Recuperado de <https://goo.su/pZRvLu2>
- Mateu, C. (2004). Desarrollo de aplicaciones Web. Editorial Eureka Media, SL. Recuperado de: <https://goo.su/SdZDnk>
- Mendoza, I (2015) realizo el proyecto de investigación “Definición de un Framework para aplicaciones Web con navegación sensible a concerns”. Realizado en Argentina. Recuperado de: <https://goo.su/Z9U2YuA>
- Mischook, S. (2007). Aprenda Programación Orientada a Objetos (POO) en PHP. Recuperado de: <https://goo.su/UnSBpN>
- Nixon, R. (2014). Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (Fourth edition). O’Reilly Media, Inc. <https://goo.su/7zf53bz>
- Pérez, H (2010) realizo el proyecto de investigación “Propuesta de Análisis y Diseño Basada en UML y UWE para la migración de arquitectura de software centralizada hacia internet”. Realizado en Guatemala. Recuperado de: <https://goo.su/iAIa>
- Pusarico, J. (2017) realizo el proyecto de investigación “Sistema web de control de seguimiento de permisos de viaje para menores de edad”. Realizado en Bolivia. Recuperado de: <https://goo.su/MrG34>
- Quispe, J. y Vílchez, J. (2017) realizo el proyecto de investigación “relación entre la implementación de un sistema de trámite documentario y la gestión documentaria de la municipalidad distrital del Rimac”. Realizado en Perú, Recuperado de <https://goo.su/P7VV>
- Ramos Cornejo, Jorge. (2020). Aplicación web para mejorar la gestión y control documental de la municipalidad distrital de Miguel Checa - Sojo [Trabajo de Investigación, Universidad Nacional, Piura] <https://goo.su/69AUZ>



- Rozo, J. (2014). Metodología de Desarrollo de Software: MBM (Metodología Basada en Modelos). Universidad Libre Barranquilla, Año 9, No. 16, pp. 111-125. ISSN: 1909-2458. Recuperado de <https://goo.su/yAORZ>
- Ruiz, Franciso. (2001). Archivística, Archivo, Documento de archivo. <https://goo.su/Avr06gW>
- Sommerville, I. (2011). Ingeniería informática (9th ed). Pearson. <https://goo.su/U9vczQ7>
- Yugcha Rocana, Delia. (2021). Sistema de gestión documental binaria para la información gerencial de la Escuela de Conducción San Miguel Drive del cantón Salcedo. [Trabajo de Investigación, Universidad Técnica de Cotopaxi] <https://goo.su/2WVCUyS>



Anexos

Anexo 1: Entrevista con el Funcionario de Informática Integrante del Departamento de Administración General y su Rol en el Área de Archivos

Para esta actividad se realizó una entrevista utilizando la herramienta de cuestionario donde se indica las necesidades de poder contar con un sistema de gestor de archivos.

<https://goo.su/jnvz>

Entrevista sobre el área de Archivos en la Gerencia Regional de Energía, Minas e Hidrocarburos de Cusco

¿Qué rol cumple el área de Archivos en la Gerencia Regional de Energía, Minas e Hidrocarburos de Cusco?

Cada área respectiva realizar un trámite esta información se almacena en el área de archivos

¿Cómo se organizan los documentos en la Gerencia Regional de Energía, Minas e Hidrocarburos de Cusco?

El área de archivos se organiza de acuerdo a las áreas que tiene, según el organigrama que tiene la institución en el área de Hidrocarburos, electrificación, minería así también el área de Administración, área de patrimonio, dentro de las áreas de proyecto se lleva las preliquidaciones, los avances físicos y financieros de cada proyecto, así también se llevan todas las resoluciones, informes, memorándum que emite la gerencia general de energía y minas de aquí de Cusco.

¿Qué tipo de documentos se manejan en el área de Documentos? Se manejan de acuerdo al tipo de áreas, años y folios, dentro de cada folio están los archivos toda esta información depende del periodo de gestión.

¿Cómo se mantiene el registro y seguimiento de los documentos que entran y salen del área de Archivos? No se cuenta con un registro de entrada y salida, cada personal que trabaja en la Gremh puede venir y recopilar información, así que cada encargado de área es responsable de la información.

¿Cómo se gestionan los documentos confidenciales en el área de Archivos? En caso de que venga contraloría ahí tendríamos que sacar información el problema es que esta en desorden y no esta digitalizado, tenerlo digital se puede evidenciar de una manera más rápida

¿Cómo se asegura el área de Archivos de que los documentos sean estén actualizados?

Mensualmente se guardan en folios y dentro de ello los archivos, en caso del área de proyectos cada fin de mes hacen su informe de avance físico y financiero una vez que esto se presenta a la Región se envía nuevamente a la Gremh y se guarda de esta manera se organiza por periodo de mes, aproximadamente en un mes se guarda 40 folios

¿Hay algún responsable que se encargue del área de Archivos? Por el momento no se cuenta con un personal responsable, solo está a cargo del área de Administración general

¿Qué medidas se toman para garantizar la privacidad y la seguridad de la información contenida en los archivos? Que dentro del almacén de archivos está cerrado y cuidado ya que hay mucha información valiosa siendo así que para poder acceder se tiene que pedir permiso al área de Administración de otra forma no se puede ingresar al área de archivos.

¿Cuál es la importancia del área de Archivos en la Gerencia Regional de Energía, Minas e Hidrocarburos de Cusco? Se ejecuta durante una gestión todo el avance, lo emitido prácticamente es el módulo más exclusivo ya que se guarda toda la información que emite cada sub gerencia en caso venga un control interno o algún tipo de fiscalización para evidencien el trabajo actual que realiza cada sub gerencia.

¿Qué procedimientos sigue el área de Archivos para mantener la seguridad de los documentos? Es un ambiente cerrado exclusivamente para archivos y esta custodiado por la vigilancia a cargo de Administración así también se comparte el ambiente con el servicio de limpieza ya que también se almacena productos de limpieza



En la **Figura 9** se evidencia cómo el Área de archivos es analizada para conocer su estructura, determinar si existe algún encargado, lo cual todavía no se ha designado por el momento. También se examina la forma en que se organiza la información, el tipo de documentos que se manejan y se nota un significativo aumento en la cantidad de información. A medida que los archivos continúan ocupando más espacio del disponible, se hace necesario digitalizar y sistematizar este proceso, que actualmente es llevado a cabo de manera manual.

Anexo 2: Presupuesto

Tabla 13

Presupuesto para el proyecto de investigación desarrollo del sistema de gestor de archivos.

Concepto	Costo Estimado (soles)
Análisis y planificación	s/.200
Diseño de la interfaz	s/.300
Desarrollo de funcionalidades	s/.400
Pruebas y depuración	s/.200
Documentación y manuales	s/.100
Contingencia	s/.100
Hosting local básico	s/.100
Aplicaciones de tecnología	s/.100
Almacenamiento de datos	s/.250 (s/.50/mes * 5 meses, 1 GB) Almacenamiento estimado para 1 GB durante 5 meses
Digitalización	s/.250 (Costo total estimado para la digitalización)
Total	s/.2000 (Suma correcta de todos los costos)



Anexo 3: Matriz de consistencia basado

Tabla 14

Matriz de consistencia del sistema gestor de archivos para la Gremh en el área de Archivos

Título: "Sistema de Gestor de Archivos basado en la tecnología Bootstrap en el Área de Archivos de la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco"				
Problema de investigación	Justificación	Objetivo general	Cronograma de actividades	Resultados esperados
¿De qué manera se implementará el sistema de gestor de archivos basado en la tecnología Bootstrap para el gestor de archivos en la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco en el área de documentos?	La necesidad de la Gremh en el área de archivos de contar con una herramienta tecnológica que permita una gestión más eficiente, segura y organizada de los documentos relacionados con las actividades de las áreas mencionadas, la implementación de un sistema de gestor de archivos permitirá optimizar la gestión	Implementar un sistema de gestor de archivos basado en la tecnología Bootstrap en el área de archivos de la Gerencia Regional de Energía, Minas e Hidrocarburos - Cusco Objetivos específicos • Identificar requisitos del usuario a través de entrevistas con usuarios y equipo de la Gerencia Regional de Energía,	Detalle de actividades Plan de trabajo de desarrollo de gestor de archivos en la Gremh (gerencia regional de energía, minas e hidrocarburos - Cusco) – Área de Documentos (https://goo.su/qQp6PGL)	• Se espera que el sistema de gestor de archivos permita una mayor eficiencia en la gestión de documentos en la Gerencia Regional de Energía, Minas e Hidrocarburos de Cusco, lo que podría resultar en una reducción significativa en los tiempos de búsqueda y recuperación de



documental, reduciendo tiempos y costos asociados a la gestión física de los documentos, garantizando, la integridad y la seguridad de la información y mejorando la gestión documental.

Minas e Hidrocarburos - Cusco.

- Diseñar y desarrollar el sistema de gestor de archivos usando la tecnología Bootstrap, siguiendo un enfoque iterativo e incremental para adaptarse a las necesidades del usuario.
- Realizar el despliegue del sistema de gestor de archivos usando la tecnología Bootstrap en la Gerencia Regional de Energía, Minas e Hidrocarburos – Cusco.

información, una mejor organización y control de los documentos y una mayor seguridad en el manejo de información sensible. Además, se espera que el sistema de gestor de archivos tenga un impacto positivo en la productividad y el desempeño de la gerencia regional en el área de documentos.