

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Sistema de Gestión de Conocimientos (SGU)

Héctor Nicolás Redrobán Narváez

Ingeniería en ciencias de la computación

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero en ciencias de la computación

Quito, 20 de diciembre de 2021

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

**HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA**

Sistema de Gestión de Conocimientos (SGU)

Héctor Nicolás Redrobán Narváez

Nombre del profesor, Título académico

Daniel Riofrío, Ph. D

Quito, 20 de diciembre de 2021

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombres y apellidos: Héctor Nicolás Redrobán Narváez

Código: 00203474

Cédula de identidad: 1720633153

Lugar y fecha: Quito, 20 de diciembre de 2021

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en <http://bit.ly/COPETHeses>.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on <http://bit.ly/COPETHeses>.

RESUMEN

Este proyecto trata de la implementación completa de un sistema de gestión de conocimientos el mismo que está adaptado a herramientas modernas como: NodeJS y Express, basado en lenguajes de programación actuales, como lo son: JavaScript, HTML5 y CSS. El objetivo principal del proyecto es brindar al usuario un control total acerca de la información que el sistema almacena. El proyecto es capaz de dar al usuario acceso a diferentes módulos de administración de la información y también de gestionar y filtrar los datos a través de un motor de búsqueda flexible y optimizado usando la herramienta ElasticSearch. Este proyecto está totalmente dirigido a un grupo en específico, por lo que es personalizado. Sin embargo, el proyecto es capaz de escalar a futuras actualizaciones con las cuales será adaptables a cualquier caso de uso que englobe un sistema de gestión de información. El proyecto está abierto a la modificación a cada uno de sus módulos y también a futuras implementaciones que agreguen funcionalidades extras al mismo.

Palabras clave: Sistema de gestión, NodeJS, JavaScript, ElasticSearch, motor de búsqueda, JWT, HTML5, MySQL.

ABSTRACT

This project deals with the complete implementation of a knowledge management system which is adapted to modern tools such as: NodeJS and Express. And based on current programming languages, such as: JavaScript, HTML5 and CSS. The main objective of the project is to give the user full control over the information that the system stores. The project can give the user access to different information management modules and to manage and filter the data through a flexible and optimized search engine using ElasticSearch. This project is totally targeted to a specific group, so it is customized. However, the project is capable of scaling to future updates with which it will be adaptable to any use case that encompasses an information management system. The project is open to modification to each of its modules and to future implementations that add extra functionality to it.

Key words: Management system, NodeJS, JavaScript, ElasticSearch, search engine, JWT, HTML5, MySQL.

TABLA DE CONTENIDO

INTRODUCCIÓN.....	10
DESARROLLO DEL TEMA.....	11
ANÁLISIS DE REQUISITOS	11
REQUISITOS FUNCIONALES.....	11
DISEÑO	13
CAPA DE BASE DE DATOS.....	13
CAPA DE VISTA.....	14
CAPA LÓGICA.....	14
IMPLEMENTACIÓN	15
IMPLEMENTACIÓN CAPA DE BASE DE DATOS.....	15
IMPLEMENTACIÓN CAPA DE VISTA (CLIENTE).....	15
IMPLEMENTACIÓN CAPA LÓGICA (SERVIDOR).....	16
IMPLEMENTACIÓN CAPA LÓGICA (MOTOR DE BÚSQUEDA).....	17
IMPLEMENTACIÓN DE GESTIÓN DE ARCHIVOS.....	18
IMPLEMENTACIÓN DE DESCARGA DE ARCHIVOS.....	19
IMPLEMENTACIÓN DE AUTENTICACIÓN DE USUARIOS.....	20
IMPLEMENTACIÓN DE NAVEGACIÓN SEGÚN ROLES EN EL SISTEMA.....	21
IMPLEMENTACIÓN DE MÓDULO DE ADMINISTRACIÓN DE USUARIOS.....	21
IMPLEMENTACION DE MÓDULO DE RECUPERACIÓN DE CONTRASEÑA.....	21
PRUEBAS	23
AÑADIR USUARIO.....	23
ADMINISTRAR USUARIO.....	26
AÑADIR PROYECTO.....	29
ADMINISTRAR PROYECTO.....	30
MOTOR DE BÚSQUEDA.....	32
CONCLUSIONES.....	34
REFERENCIAS BIBLIOGRÁFICAS	35
ANEXO A: PROTOTIPO DISEÑO INTERFAZ	36

ÍNDICE DE TABLAS

TABLA 1: MOTOR DE BASE DE DATOS IMPLEMENTADO EN LA SOLUCIÓN	15
TABLA 2: LENGUAJE Y LIBRERÍAS UTILIZADAS PARA LA IMPLEMENTACIÓN DE LA CAPA DE VISTA (CLIENTE).....	16
TABLA 3: LENGUAJE Y LIBRERÍAS IMPLEMENTADA EN CAPA LÓGICA (SERVIDOR).....	17
TABLA 4: LENGUAJE Y LIBRERÍAS IMPLEMENTADA EN CAPA LÓGICA (MOTOR DE BÚSQUEDA).....	18

ÍNDICE DE FIGURAS

FIGURA 1: MODELO DE MYSQL WORKBENCH (BASE DE DATOS SGC) ..14

**FIGURA 2: ESTRUCTURA DE LOS DIRECTORIOS PARA GESTIONAR
PROYECTOS.....18**

**FIGURA 3: ESTRUCTURA DE LOS DIRECTORIOS PARA GESTIONAR
ARCHIVOS TEMPORALES.19**

INTRODUCCIÓN

El grupo de investigación de la USFQ, ASP & ML (Applied Signal Processing and Machine Learning), actualmente está almacenando todos los datos de investigación de sus miembros en un repositorio no automatizado. Así mismo, el grupo buscó implementar una solución que les permita aprovechar al máximo la información reunida por todos los miembros. Se implementó una aplicación web basada en los requerimientos del grupo. Actualmente, la solución permite administrar los proyectos del grupo mediante una interfaz que permite gestionar toda la información acerca del proyecto. Así mismo, la solución permite administrar a sus usuarios y clasificarlos por roles. El proyecto, además de ofrecer una administración y gestión de sus entidades (proyectos y usuarios), tiene la capacidad de filtrar información entre todos los proyectos almacenados, apuntando así a un mejor manejo de los datos reunidos. Una característica adicional que ofrece la solución es actuar como un medio de acceso a la información publicada para cualquier persona que tenga interés. Hablando en términos de ingeniería en ciencias de la computación, la solución usa herramientas que engloban el área tales como: base de datos, seguridad informática, aplicaciones distribuidas, minería de datos y las tecnologías emergentes actuales.

DESARROLLO DEL TEMA

ANÁLISIS DE REQUISITOS

Se tomó en cuenta varios requisitos los cuáles permitieron definir las diferentes funcionalidades de la solución.

REQUISITOS FUNCIONALES

1. INTERFAZ GRÁFICA

El principal requisito de la interfaz gráfica es que debe ser intuitiva para el usuario y amigable al momento de indexar datos mediante los proyectos, por lo que se realizó una página de inicio, un menú principal, una página de administración de usuarios, una página de administración de proyectos y un filtrado de información.

El prototipo base del diseño se encuentra en el Anexo A.

El diseño final se apega a las especificaciones descritas en el manual de marca de la USFQ, en especial los colores y la tipografía.

2. BASE DE DATOS

Los requerimientos para estructurar la base de datos se modificaron en las fases iniciales del proyecto.

Para una primera versión de la base de datos se realizó un análisis a partir de entidades. Se llegó a la conclusión de que el sistema trabajará con las entidades: usuario, proyecto, colaboradores e institución.

Usuario: hace referencia a un investigador sin importar el rol que este tenga.

Proyecto: hace referencia a un proyecto de investigación del grupo abarcando todas sus partes.

Colaboradores: es una unión de las entidades usuario y proyecto la cual hace referencia a todos los colaboradores dentro de un proyecto.

Institución: hace referencia a cualquier institución en convenio con el grupo de investigación.

3. IDENTIFICACIÓN DE USUARIOS

La identificación del usuario es un requisito fundamental para gestionar los niveles de acceso en la solución.

Se requirió la administración del nivel de acceso de un usuario mediante roles. Se contempló la existencia de 2 roles los cuales tienen acceso a diferentes funcionalidades. Se buscó que los roles se ajusten a las necesidades del grupo y se analizó la creación de diferentes tablas y menús que cambian de acuerdo con el usuario en cuestión.

El manejo de roles es capaz de asignar varios niveles de autoridad en la plataforma.

4. GESTIÓN DE USUARIOS

El módulo de gestión de usuarios se analizó de tal manera que, dependiendo del rol, el usuario del sistema acceda únicamente a la información permitida para su rol.

El módulo en base al análisis es versátil y entendible para los usuarios que manejarán el mismo.

5. GESTIÓN DE PROYECTOS

La gestión de proyectos se basó dependiendo de los roles. Los usuarios administradores van a ser capaces de crear proyectos y editar los mismos. Cada proyecto cuenta con sus colaboradores los cuales van a gestionar de manera

limitada los proyectos.

6. MOTOR DE BÚSQUEDA

Para el módulo de motor de búsqueda se escogieron tres filtros los cuales permitirán al usuario filtrar la información de los proyectos de una manera específica. Este motor está disponible tanto para usuario dentro de la plataforma (administradores e investigadores), como para público en general restringiendo parámetros de búsqueda.

DISEÑO

A través del análisis anterior se diseñó la aplicación en 3 diferentes capas las cuales son: la capa de vista, la capa lógica y la capa de base de datos.

CAPA DE BASE DE DATOS

Se diseñó un modelo entidad-relación, en la herramienta MySQL Workbench (ver Figura 1), donde se planteó las 4 clases analizadas previamente con sus atributos necesarios y la relación entre ellas. Las entidades principales son usuario y proyecto. La entidad usuario tiene como atributos: nombre, apellido, email, salt, hash, rol y id_usuario como su llave principal. La entidad proyecto tiene como atributos: título, descripción, estado, url_overleaf, url_github, paper, script, responsable y id_proyecto como su clave principal. La entidad institución tiene como atributos: nombre y país. Se creó una relación muchos a muchos entre las entidades usuario y proyecto lo cuál nos generó una nueva entidad, la entidad colaboradores la cual tiene como atributos: id_usuario y id_proyecto, las cuales son las llaves principales de las entidades usuario

y proyecto.

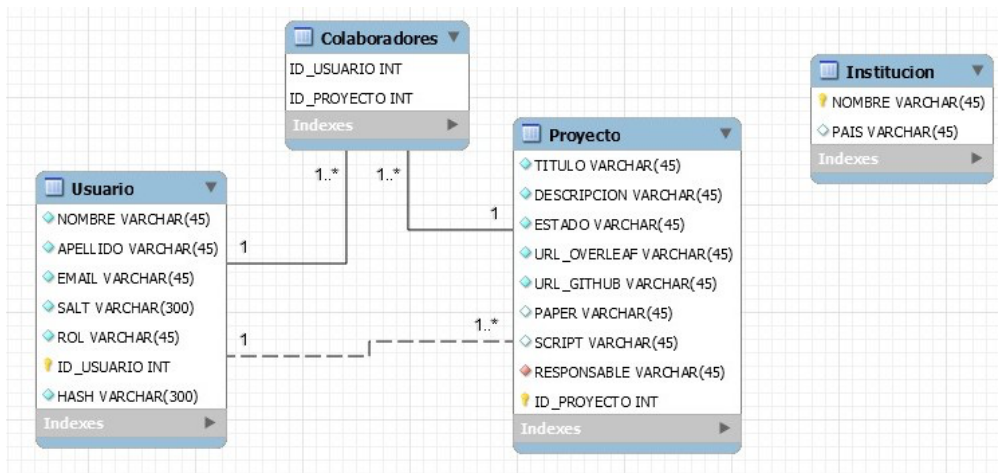


Figura 1: Modelo de MySQL Workbench (Base de datos SGC)

CAPA DE VISTA

La capa de vista es basada en el prototipo presentado en la fase de requerimientos y fue diseñada en la herramienta Figma, la cual es una herramienta abierta al público para crear diseños para cualquier plataforma. Toda la implementación de la solución se apejó a este modelo inicial el cual está disponible en los Anexos. (Ver Anexo A)

CAPA LÓGICA

Esta capa establece todo lo que a funcionalidad se refiere y está basada en los requerimientos iniciales. Mediante la información proporcionada por el usuario final se logró crear la lógica que tiene el sistema, tanto en tema de control y flujo de datos como en interfaces gráficas. La capa lógica está constituida por el Modelo Vista Controlador (MVC) la cual separa y organiza cada fragmento del proyecto.

Adicionalmente, la capa lógica esta constituida por el servidor del motor de búsqueda la cual se especifica en el apartado de implementación de capa lógica (motor de búsqueda). Este fragmento de la capa lógica permite a la aplicación tener una réplica de la información dentro de la base de datos y utilizando algoritmo de procesamiento sobre los datos enviar información de filtrado hacia el servidor.

IMPLEMENTACIÓN

Con respecto a lo mencionado en el apartado de diseño, el mismo se implementó con las siguientes tecnologías.

IMPLEMENTACIÓN CAPA DE BASE DE DATOS

La base de datos se implementó siguiendo el paradigma de diseño de bases de datos actuales, así como las reglas de normalización. Cada entidad creada para solucionar el problema en cuestión sigue el modelo entidad relación para su funcionamiento. La tabla 1 muestra el motor de la base de datos utilizada y su versión.

Motor base de datos	Versión
MySQL	8.0

Tabla 1: Motor de base de datos implementado en la solución

IMPLEMENTACIÓN CAPA DE VISTA (CLIENTE)

Para la implementación de la capa visual se usaron tecnologías emergentes las cuales nos permiten integrar varias herramientas que ayudan a la comunicación de cada capa. Esta capa está construida en base al lenguaje PUG el cual es interpretado por HTML 5 y finalmente compilado por el navegador. La funcionalidad que permite a la solución no actuar de forma estática está respaldado por la tecnología de JavaScript para el manejo dinámico de los elementos HTML y JQuery para dar una funcionalidad más volátil a los elementos que se comunican con la capa de servidor. La tabla 2 muestra las versiones de las librerías utilizadas asociadas con cada lenguaje de programación.

Lenguajes	Librerías
JavaScript	JQuery 3.6.0 jQuery UI 1.13.0
HTML 5	Pug 3.0.2 Bootstrap 5.1.1 Bootstrap-table 1.18.3 Fontawesome CSS

Tabla 2: Lenguaje y librerías utilizadas para la implementación de la capa de vista

(Cliente)

IMPLEMENTACIÓN CAPA LÓGICA (SERVIDOR)

El servidor está basado en el lenguaje JavaScript para el funcionamiento de todas las herramientas y NodeJS para toda la estructura del servidor. Varias librerías de NodeJS son utilizadas para comunicarse con el cliente, así como dar respuestas al cliente en base a las peticiones procesadas por la base de datos. El sistema de autenticación y sistema de navegación dependen también de NodeJS y sus librerías para su funcionamiento. La tabla 3 muestra las versiones de las librerías utilizadas asociadas con cada lenguaje de programación.

Lenguajes	Librerías
NodeJS	Node 14.15.5 npm 6.14.11 axios 0.21.4 cookie-parser 1.4.4 debug 2.6.9 dotenv 10.0.0 express 4.16.1 http-errors 1.6.3 jsonwebtoken 8.5.1 Morgan 1.9.1 Mysql 2 2.3.0 nodemailer 6.7.0 passport 0.5.0 passport-jwt 4.0.0 pug 3.0.2

Tabla 3: Lenguaje y librerías implementada en capa lógica (Servidor)

IMPLEMENTACIÓN CAPA LÓGICA (MOTOR DE BÚSQUEDA)

INTRODUCCIÓN A ELASTICSEARCH

El motor de búsqueda se ejecuta a través de un servidor adicional el cual recibe peticiones HTTP y responde peticiones complejas sobre la información de los proyectos. Elasticsearch es según la documentación oficial:

“Un motor de analítica y análisis distribuido gratuito y abierto para todos los tipos de datos, desarrollado a partir de ApacheLucene y es conocido por sus API REST simples, velocidad y escalabilidad.”

¿POR QUÉ USAR ELASTICSEARCH?

Elasticsearch es una herramienta que fue lanzada en 2010 por lo que a lo largo del tiempo se ha ido perfeccionando y actualmente es el motor más usado para indexación de datos. Debido a su escalabilidad y velocidad elasticsearch es capaz de indexar múltiples tipos de contenido, siendo apto para diferentes casos de usos.

¿COMÓ FUNCIONA ELASTICSEARCH?

Elasticsearch crea copias de la información en nuestra base de datos y las almacena

en un índice el cual es una colección de documentos relaciones entre sí, los datos sin procesar fluyen hacia el servidor de elasticsearch y antes de ser indexados estos pasan por un proceso de preprocesamiento mediante el cual los datos son normalizados y enriquecidos antes de su indexación. Una vez indexados los usuarios pueden hacer consultas complejas sobre sus datos hacia el servidor para obtener un resumen complejo de los mismos. El servidor de elasticsearch se ejecuta por defecto en el puerto 9200.

Lenguajes	Librerías
NodeJS	elasticsearch 16.7.2 pdf-parse 1.1.1 express-fileupload 1.2.1

Tabla 4: Lenguaje y librerías implementada en capa lógica (Motor de búsqueda)

IMPLEMENTACIÓN DE GESTIÓN DE ARCHIVOS

El sistema consta de dos metodologías para manejar los archivos, estas ayudan al funcionamiento del módulo de administración de proyectos y el módulo del motor de búsqueda.

MANEJO DE ARCHIVOS DE UN PROYECTO

Para gestionar la información de un proyecto se manejó directorios los cuales contienen el directorio principal y 2 subdirectorios los cuales contendrán los anexos del proyecto. Los identificadores de cada directorio lucen de la siguiente manera, ver figura 2.

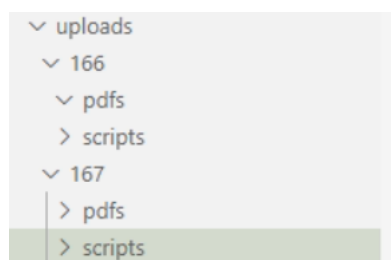


Figura 2: Estructura de los directorios para gestionar proyectos.

La figura 2 muestra como está estructurada la gestión de los proyectos a nivel de carpetas, la carpeta raíz tiene el nombre de “uploads” y cada subdirectorio hace referencia a un proyecto en específico el cual tiene como identificador el ID del proyecto en la base de datos. Cada subdirectorio de un proyecto hace referencia a los archivos adjuntos que pueda tener el mismo (papers y scripts).

MANEJO DE ARCHIVOS TEMPORALES

Los archivos temporales tienen la función de almacenar únicamente archivos de descarga, esto quiere decir que cuando un usuario intente descargar cualquier archivo adjunto de un proyecto este se almacenará en una carpeta temporal dentro del sistema para poder ser descargado y tiene la siguiente estructura, ver figura 3.

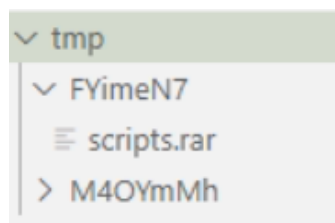


Figura 3: Estructura de los directorios para gestionar archivos temporales. La figura 3 muestra la estructura del directorio de archivos temporales, cada subdirectorio dentro del directorio “tmp” contiene archivos temporales los cuales podrán ser descargados y el nombre de cada subdirectorio está conformado por una cadena de 7 caracteres aleatorios.

IMPLEMENTACIÓN DE DESCARGA DE ARCHIVOS

Los archivos descargables son cualquier adjunto de un proyecto, para poder realizar la descarga de un archivo adjunto cuando un usuario lo requiere el servidor identifica sobre que sistema operativo se está ejecutando y utiliza los comandos de compresión dependiendo del mismo, después localiza el o los archivos en cuestión y utiliza procesos del sistema para comprimir todos los archivos en un solo documento.

Finalmente, el sistema mueve este archivo al directorio de archivos temporales y es desde aquí de donde podrá ser descargado por el usuario en el formato reconocido por el sistema.

Para sistemas operativos Windows el sistema requiere que se tenga instalada la aplicación WinRAR y que esté añadida a las variables de entorno. Para sistema operativo Linux y MacOS se utiliza la herramienta zip instalada por defecto.

IMPLEMENTACIÓN DE AUTENTICACIÓN DE USUARIOS

La solución integra una arquitectura de autenticación de usuarios capaz de transferir la identidad de cada usuario entre los diferentes módulos. Facilitando así que los usuarios no tengan que ser autenticados una y otra vez.

La autenticación de usuarios está basada en criptografía y JWT. Según la página oficial de JSON Web Token:

“JSON Web Token (JWT) es un estándar abierto (RFC 7519¹) que define una vía compacta para transmitir información de forma segura entre vínculos como objetos JSON. Esta información puede ser verificada y es confiable debido a que digitalmente firmada, su estructura consiste en un objeto JSON² compuesto por 3 atributos (header, payload, signature)”*

Se utilizó estos dos paradigmas para autenticar a los usuarios mediante la creación de token, el cual tendrá como payload el ID del usuario, será tipo JWT y se firmará mediante una clave pública y una clave privada, las cuales se crearon con el algoritmo RS256. El administrador tiene que registrar al usuario utilizando todos los campos requeridos y el sistema como medida de precaución encripta la contraseña usando la estrategia hash/salt.

¹ RFC 7519: estándar definido para la arquitectura JWT.

² Objetos JSON: formato de datos basado en texto.

El usuario que intente ser autenticado en el sitio web pasará por el módulo de autenticación verificando la contraseña ingresada con su hash y su salt, si la contraseña es correcta el sistema crea un JWT único para el usuario usando la clave privada y la clave pública y lo almacena en una cookie oculta que sirve para autenticar al usuario cada vez que este necesita entrar a una ruta protegida, siempre y cuando el JWT no haya expirado.

El JWT creado por el sistema tiene una duración de 4 horas, una vez finalizado el tiempo de duración este se elimina y el usuario tendrá que ser autenticado nuevamente.

IMPLEMENTACIÓN DE NAVEGACIÓN SEGÚN ROLES EN EL SISTEMA

La navegación para el usuario se implementó de tal manera que el sistema sea capaz de reconocer que tipo de usuario está navegando por el sistema mediante el rol de este.

Dependiendo del rol que el usuario tenga el sistema responde a los privilegios a los que el usuario tiene acceso.

IMPLEMENTACIÓN DE MÓDULO DE ADMINISTRACIÓN DE USUARIOS

El módulo de administración de usuarios fue construido de tal manera que los usuarios ADMINISTRADORES puedan manejar toda la información pública de los usuarios que se encuentran registrados en el sistema. Este módulo permite editar, borrar, filtrar toda la información de un usuario en cuestión. El usuario ADMINISTRADOR también tiene la opción de quitar privilegios a los usuarios registrados.

IMPLEMENTACION DE MÓDULO DE RECUPERACIÓN DE CONTRASEÑA

El módulo de recuperación de contraseña se encuentra en la pantalla de inicio de sesión y consiste en una verificación de correo electrónico, esta verificación se basa en encontrar al usuario en la base de datos, si el usuario con el correo dispuesto existe, el

sistema genera un enlace de recuperación de contraseña el cual es enviado al correo electrónico del usuario. Este enlace incluye un token usando la tecnología JWT usando anteriormente el cual expira dentro de 1 hora. Desde este enlace el usuario puede actualizar su contraseña en caso de olvido de esta.

PRUEBAS

Las siguientes pruebas describen todos los procesos/módulos utilizados en el sistema, cada apartado de esta sección está descrita de tal manera que se pueda entender que es lo que hace cada uno de los procesos/módulos.

AÑADIR USUARIO

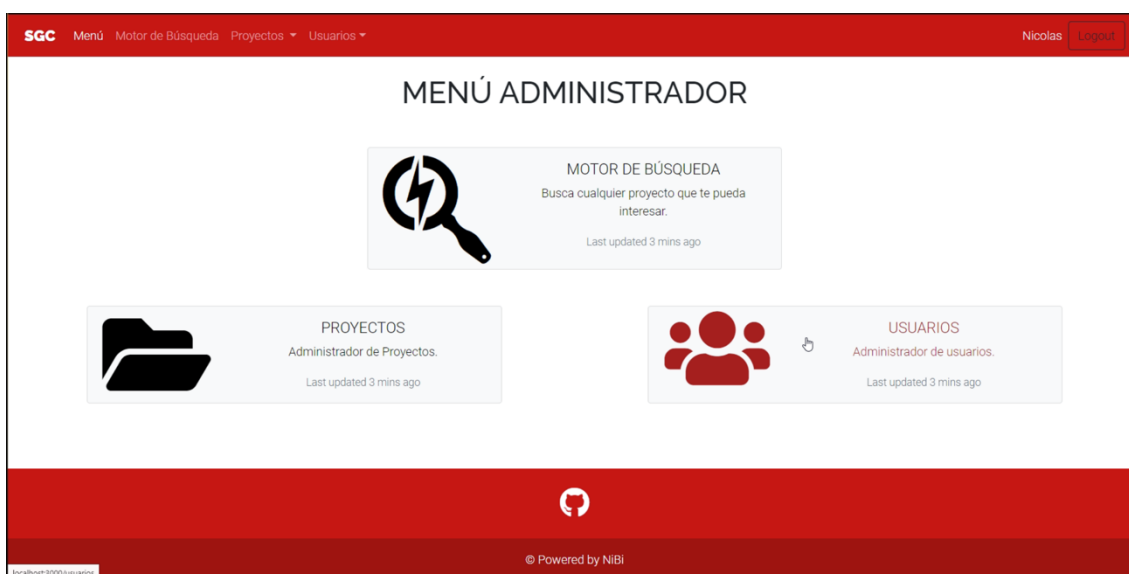


Imagen 1: Ingresar a módulo de administrador de usuarios.

La imagen 1 muestra la pantalla inicial del usuario ADMINISTRADOR con sus respectivas opciones (módulos) disponibles para navegar en el sistema. Así como un NAVBAR principal para mostrar opciones a través de todos los módulos y también la opción de iniciar sesión o cerrar sesión. Para el caso de agregar un nuevo usuario se tiene que hacer clic en el módulo de USUARIOS como se muestra en la imagen 1.

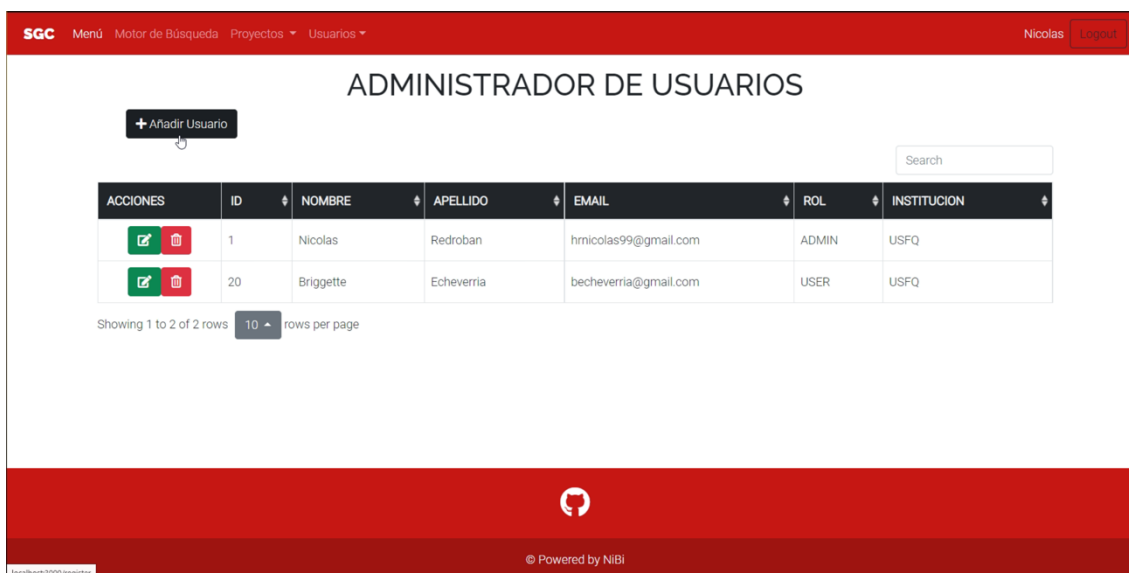


Imagen 2: Ingresar a módulo de añadir usuario.

La imagen 2 muestra el segundo paso para agregar un nuevo usuario, para proceder se deber hacer clic en el botón de “Añadir Usuario”.

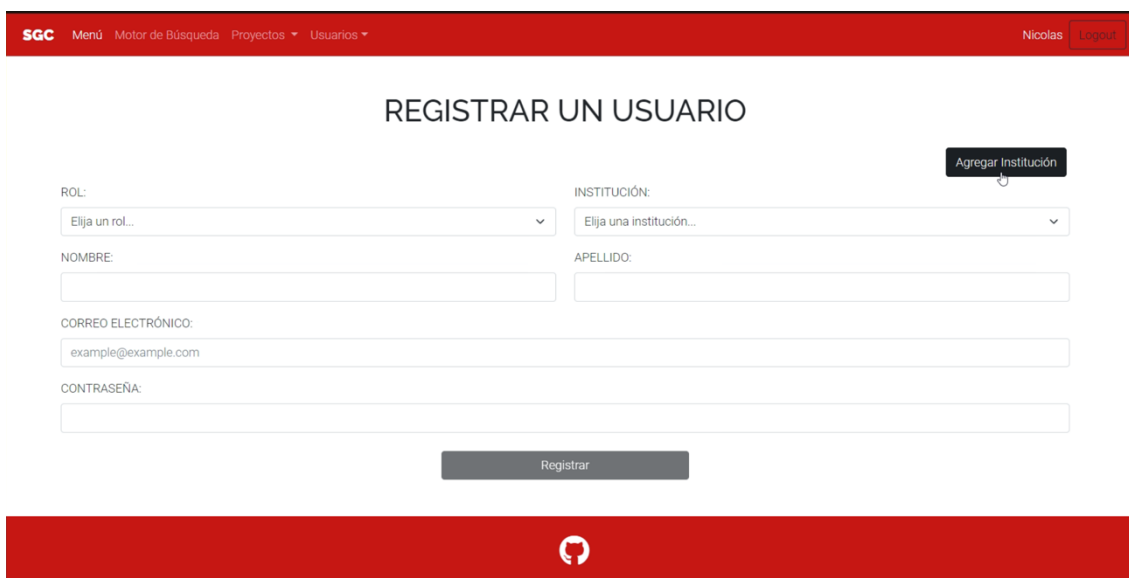
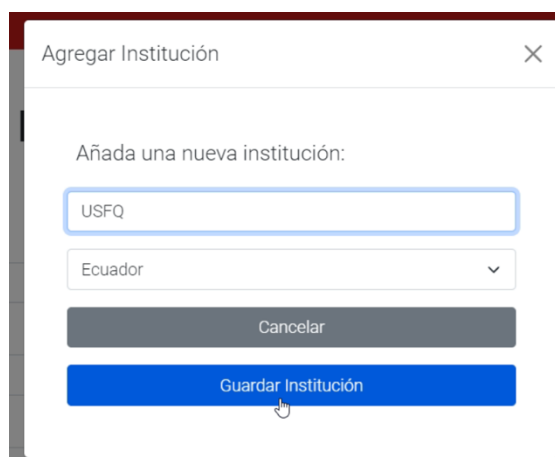


Imagen 3: Ingresar a módulo de agregar institución (opcional)

La imagen 3 muestra la pantalla para registrar un usuario a la cual solamente tienen acceso usuarios ADMINISTRADORES, desde esta pantalla el usuario también tiene la opción de agregar una nueva institución en caso de que la deseada no se encuentre disponible en la base de datos de instituciones. El usuario deberá llenar todos los campos ya que son obligatorios. La contraseña ingresada por el usuario no se

almacenará directamente en la base de datos, sino que se realizará una técnica de encriptación de la contraseña para guardar la misma y que no sea accesible fácilmente por algún ataque, tal como se explica en el apartado de implementación de autenticación de usuarios.



A screenshot of a web application dialog box titled "Agregar Institución". The dialog contains the text "Añada una nueva institución:" followed by a text input field containing "USFQ", a dropdown menu showing "Ecuador", a "Cancelar" button, and a "Guardar Institución" button. A mouse cursor is pointing at the "Guardar Institución" button.

Imagen 4: Agregar institución (opcional)

La imagen 4 muestra la funcionalidad mencionada en el apartado anterior desde la cual el usuario puede agregar una nueva institución a la base de datos, ya que al registrar un usuario nuevo este requiere pertenecer a una institución dentro de la base de datos del sistema.

SGC Menú Motor de Búsqueda Proyectos Usuarios Nicolás Logout

REGISTRAR UN USUARIO

[Agregar Institución](#)

ROL: ADMINISTRADOR INSTITUCIÓN: USFQ

NOMBRE: Daniel APELLIDO: Riofrio

CORREO ELECTRÓNICO: driofrio@usfq.com.ec

CONTRASEÑA:

[Registrar](#)

Imagen 5: Llenar campos de formulario y registrar usuario.

La imagen 5 muestra el formulario de registro de un usuario completado en su totalidad listo para ser guardado en la base de datos.

ADMINISTRAR USUARIO



Imagen 6: Ingresar a módulo de administrador de usuarios.

La imagen 6 muestra el primer paso para administrar un usuario, el ADMINISTRADOR tiene que hacer clic en el módulo USUARIOS para poder realizar cualquier modificación.

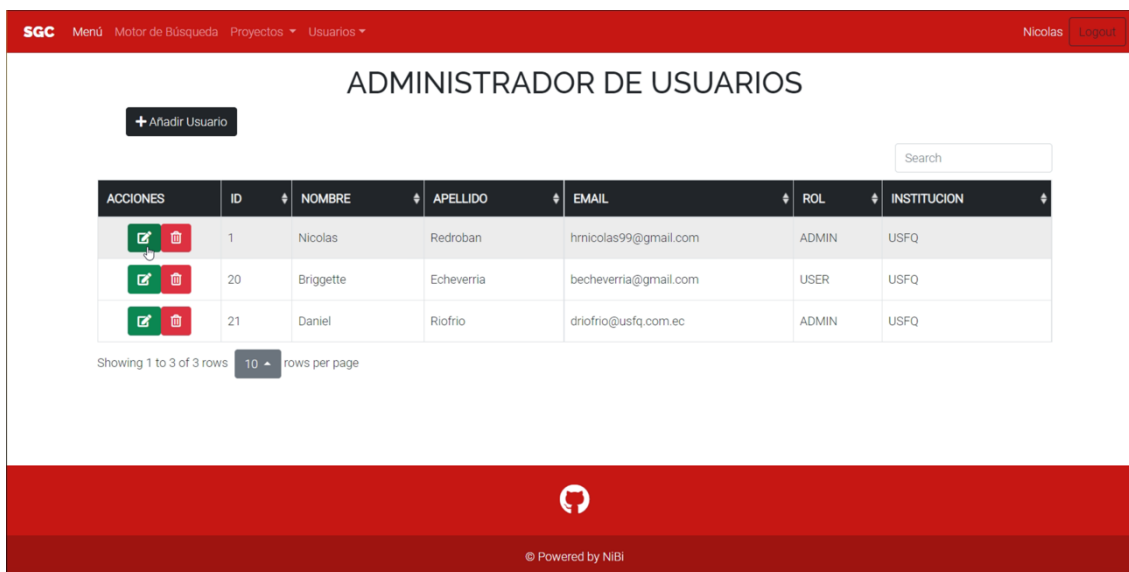








Imagen 7: Ingresar a módulo para editar usuario (botón verde)

La imagen 7 muestra la pantalla desde la cual el usuario ADMINISTRADOR podrá editar, añadir y deshabilitar usuarios. La opción de eliminar un usuario estará disponible solamente si éste no tiene ninguna dependencia en cualquier proyecto sin importar que éste se encuentre en estado INACTIVO. Para comenzar a editar un usuario se debe hacer clic en el botón verde del usuario correspondiente.

Imagen 8: Editar información y guardar.

La imagen 8 muestra la pantalla que se despliega después de haber hecho clic en el botón verde del usuario que se desea modificar. Desde esta pantalla el ADMINISTRADOR puede editar cualquier información del usuario a excepción de la contraseña debido a que el ADMINISTRADOR no puede modificar la contraseña después de que el usuario es creado, en su defecto el usuario puede modificar su contraseña usando el módulo de recuperación de contraseña tal y como se explica en el apartado de implementación de módulo de recuperación de contraseña.

The screenshot shows the 'ADMINISTRADOR DE USUARIOS' interface. At the top, there is a red navigation bar with 'SGC' and links for 'Menú', 'Motor de Búsqueda', 'Proyectos', and 'Usuarios'. On the right, the user 'Nicolas' is logged in. The main content area is titled 'ADMINISTRADOR DE USUARIOS' and features a '+ Añadir Usuario' button and a search bar. Below this is a table with the following data:

ACCIONES	ID	NOMBRE	APELLIDO	EMAIL	ROL	INSTITUCION
 	1	Nicolas	Redroban	hmicolas99@gmail.com	ADMIN	USFQ
 	20	Briggette	Echeverria	becheverria@gmail.com	USER	USFQ
 	21	Daniel	Riofrio	driofrio@usfq.com.ec	ADMIN	USFQ

Below the table, it indicates 'Showing 1 to 3 of 3 rows' and '10 rows per page'. The footer of the page is red and contains a logo and the text '© Powered by NIBI'.

Imagen 9: Ingresar a módulo para eliminar usuario (botón rojo)

La imagen 9 muestra el primer paso para eliminar un usuario. Para proceder a eliminar un usuario se debe hacer clic en el botón rojo del usuario correspondiente. Es importante recalcar que un usuario podrá ser eliminado únicamente si el mismo no es dependiente de ningún proyecto, es decir, si el usuario en cuestión no se encuentra incluido en algún proyecto dentro del sistema.

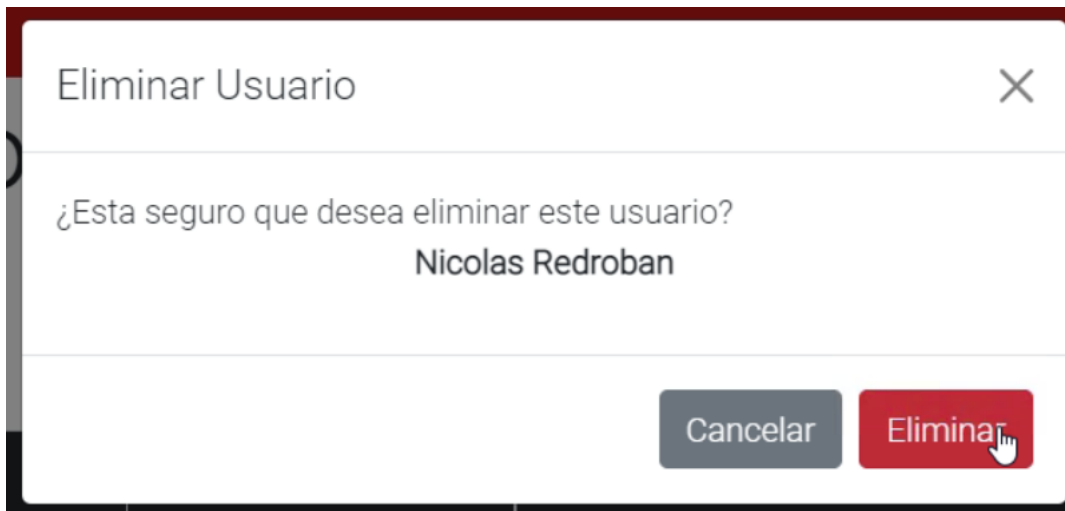


Imagen 10: Confirmar y eliminar.

La imagen 10 muestra una pantalla de confirmación de eliminación la cual se muestra después de haber hecho clic en el botón rojo del usuario en cuestión. Esta pantalla sirve como medida de precaución en caso de que el ADMINISTRADOR haga clic sin intención de eliminar a un usuario.

AÑADIR PROYECTO

The image displays four sequential screenshots of the 'REGISTRAR UN PROYECTO' (Register a Project) form in a web application. Each screenshot shows a different step of the registration process.

- Step 1:** 'PASO 1: Ingresa título y estado de tu proyecto'. Fields include 'TÍTULO(*)' (Automatic ladybird beetle detection using deep-learning models), 'ESTADO(*)' (Desarrollo), 'RESPONSABLE(*)' (Nicolas Redroban), and a 'Paso 2' button.
- Step 2:** 'PASO 2: Ingresa la información principal de tu proyecto'. Fields include 'DESCRIPCIÓN (Máximo 500 caracteres(*)', 'PALABRAS CLAVES (Separar por comas | UTF-8, Data mining(*)', and a 'Paso 3' button.
- Step 3:** 'PASO 3: Ingresa links de tu proyecto'. Fields include 'URL Overleaf(*)' (https://www.overleaf.com), 'URL GitHub', and a 'Paso 4' button.
- Final Screen:** 'Se ha creado correctamente tu proyecto' with a green checkmark. The project title is 'Automatic ladybird beetle detection using deep-learning models'. It lists the collaborator 'Daniel Rodro - ddrodro@ufla.com.ec' and includes buttons for 'Añadir Colaboradores', 'Regresar al Menu', and 'Añadir nuevo Proyecto'.

Imagen 11: Formularios para registrar un proyecto.

La imagen 11 muestra el proceso por el cual un usuario tiene que pasar para poder registrar un nuevo proyecto en el sistema. Como primera pantalla el usuario puede dar un título a su proyecto el cual es verificado por el sistema en la base de datos evitando así duplicación de títulos. Después de haber agregado un título a su proyecto el sistema presenta una pantalla desde la cual el usuario agrega una descripción y palabras clave del proyecto en cuestión, una vez realizado esto el sistema muestra la última pantalla desde la cual el usuario puede agregar los enlaces desde los cuales se pueden acceder a la información del proyecto registrada en otros sitios web.

ADMINISTRAR PROYECTO

The screenshot shows the 'ADMINISTRADOR DE PROYECTOS' interface. At the top, there is a red navigation bar with 'SGC', 'Menú', 'Motor de Búsqueda', 'Proyectos', and 'Usuarios'. On the right, it shows 'Nicolas' and a 'Logout' button. Below the header, the title 'ADMINISTRADOR DE PROYECTOS' is centered. A '+ Añadir Proyecto' button is on the left, and a search bar is on the right. The main content is a table with the following structure:

ACCIONES	TÍTULO	ESTADO	RESPONSABLE
	Toward Real-Time Volcano Seismic Events' Classification: A New Approach Using Mathematical Morphology and Similarity Criteria	Desarrollo	Nicolas Redroban
DESCRIPCIÓN	PALABRAS CLAVE	COLABORADORES	LINKS
This work proposes a new approach based on a suit combination of mathematical morphology and similarity criteria techniques to classify long-period and volcano-tectonic seismic events of the Cotopaxi volcano. The proposed method explores the seismic signal domain to compute a new feature space based on the edges map of the seismic events pattern represented in the gray-level spectrogram images, which is used to feed a set of similarity-based classifiers. The L2-norm was selected as the best metric to be implemented by the proposed method. In terms of	USFQ, DATA MINING	Daniel Riofrio drioefrio@usfq.com.ec	

Imagen 12: Vista previa de un proyecto.

La imagen 12 muestra la pantalla de administración de proyectos desde la cual el usuario puede agregar, modificar o eliminar un proyecto. Es importante recalcar que el usuario simplemente tendrá acceso total a los proyectos los cuales el usuario es el responsable, en caso de que el usuario sea colaborador de un proyecto éste puede editar todos los campos con excepción del título y los colaboradores, en caso de que el usuario

no pertenezca de ninguna forma a un proyecto este no tendrá acceso alguno al proyecto simplemente una vista previa del mismo.

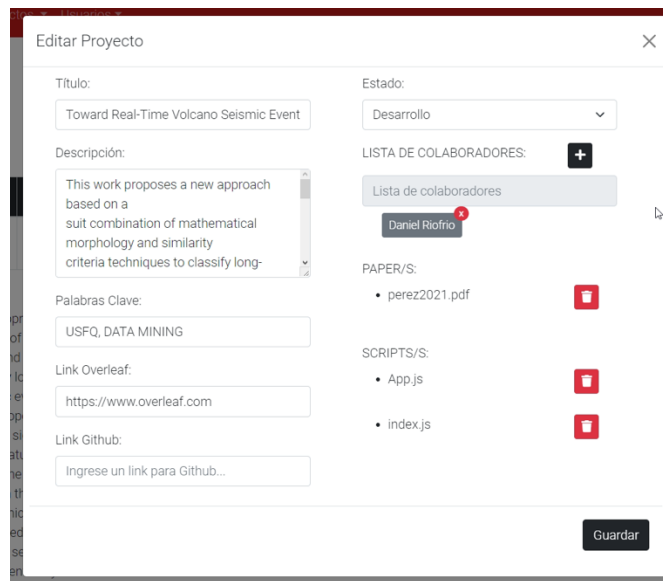


Imagen 13: Editar un proyecto (Botón verde).

La imagen 13 muestra la pantalla que el sistema muestra cuando un usuario quiera editar un proyecto. El sistema permite editar cualquier información del proyecto respetando los accesos descritos en el apartado anterior. Ver descripción de imagen 12.



Imagen 14: Módulo para cargar papers/scripts (Botón azul/amarillo).

La imagen 14 muestra la pantalla que se despliega una vez hecho clic en cualquiera de los botones azul o amarillo. El usuario puede desde esta pantalla agregar archivos relacionados con el proyecto (papers o scripts).



Imagen 15: Confirmar y eliminar un proyecto.

La imagen 15 muestra la pantalla que el sistema despliega una vez hecho clic el botón rojo de un proyecto en el administrador de proyectos. Esta pantalla sirve como medida de precaución en caso de que el usuario haya hecho clic erróneamente.

MOTOR DE BÚSQUEDA

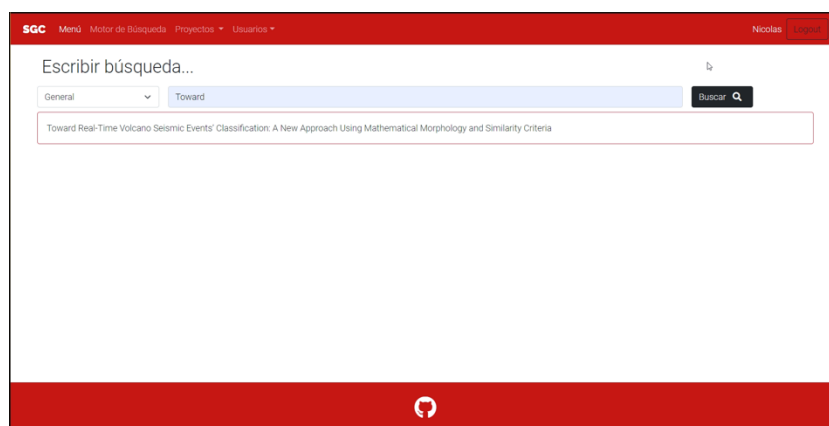


Imagen 16: Búsqueda general de un proyecto.

La imagen 16 muestra el primer paso para comenzar a usar el módulo de motor de búsqueda, esta pantalla muestra un buscador y varios filtros que se pueden utilizar para encontrar a un proyecto. Una vez escogido el filtro y escrito en el buscador el usuario puede realizar su búsqueda haciendo clic en el botón “Buscar” y el sistema mediante la implementación descrita en el apartado de implementación de motor de búsqueda,

muestra la información encontrada en la base de datos de Elasticsearch. Los resultados se presentan en modo lista siendo el primero de la lista el resultado con más relevancia según los criterios de búsqueda ingresados por el usuario.

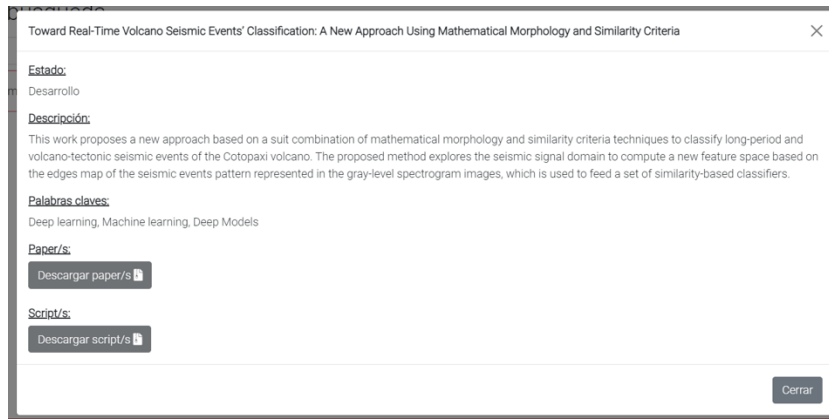


Imagen 17: Información del proyecto y descarga de archivos.

La imagen 17 muestra la pantalla que el sistema despliega una vez hecho clic en algún resultado mostrado por el motor de búsqueda. Desde esta pantalla el usuario puede visualizar una vista previa del proyecto en cuestión, así como descargar la información adjunta que tiene el proyecto (papers/scripts).

CONCLUSIONES

El proyecto detallado y redactado en este documento es una herramienta útil para el aprendizaje de nuevas tecnologías y de como integrarlas formando un producto final sólido y escalable dentro del campo del desarrollo web y ciencias de la computación. Este trabajo agrupa todos los enfoques técnicos de cada área. De este trabajo se ha reforzado exponencialmente las metodologías aprendidas a lo largo de la carrera de ciencias de la computación, estas metodologías incluyen arquitecturas como: base de datos, minería de datos, diseño de interfaces web, desarrollo de software, arquitectura de software, entre otros. A lo largo del desarrollo del trabajo se investigó temas complejos no mencionados, como las regulaciones para autenticación automática y sus aplicaciones, manejo de sistemas operativos, integración de diferentes servidores y clientes web e integración de bases de datos mediante API Rest para tecnologías emergentes como NodeJS. Todos los temas mencionados anteriormente son de vital importancia para entender en su totalidad el funcionamiento del proyecto y se recomienda investigar a fondo y prever la posible integración de futuras herramientas al proyecto. Algunas de las dificultades más relevantes en el desarrollo del proyecto fueron: integración de la base de datos mediante una API Rest, manejo de directorios a nivel de aplicación y sistema operativo, manejo de roles de usuarios y funcionamiento óptimo de motor de búsqueda.

REFERENCIAS BIBLIOGRÁFICAS

¿Qué es Elasticsearch?. Elastic. (2021). Retrieved 3 December 2021, from <https://www.elastic.co/es/what-is/elasticsearch>.

Figma: the collaborative interface design tool. (2021). Figma.
<https://www.figma.com>

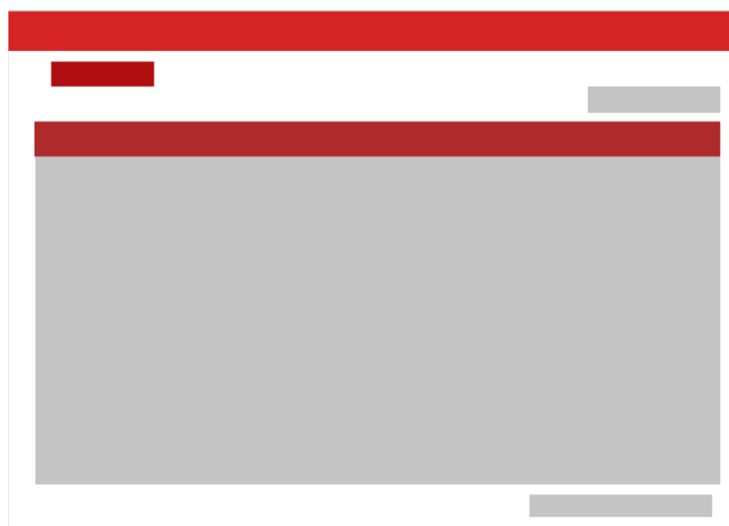
JWT.IO - JSON Web Tokens Introduction. (2021). Retrieved 15 November 2021, from <https://jwt.io/introduction>

MARKETING DIGITAL USFQ. (2021). MANUAL DE ELEMENTOS GRÁFICOS [PDF]. Revisado 10 Octubre 2021, from <https://www.usfq.edu.ec/sites/default/files/2021-09/manual-de-marca-digital.pdf>.

Node.js. Node.js. (2021). Retrieved 3 December 2021, from <https://nodejs.org/es/>.

ANEXO A: PROTOTIPO DISEÑO INTERFAZ

Este anexo representa el prototipo gráfico presentado inicialmente sobre el cual se desarrolló todo el sistema. Este prototipo fue aprobado y explicado en la presentación virtual inicial. Las imágenes descritas en los siguientes apartados sirvieron como referencia para todo el diseño del sistema tanto en la implementación lógica como en la implementación gráfica del mismo.



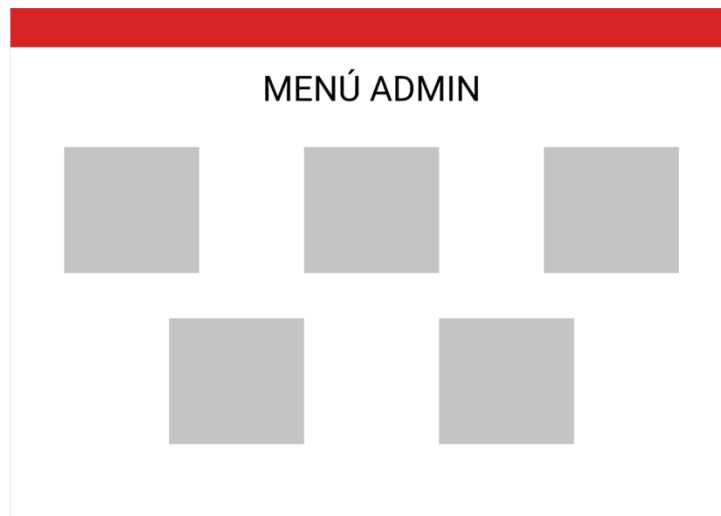
Anexo A.1: Prototipo de la interfaz para administrar proyectos/usuarios.

El Anexo A.1 muestra el prototipo presentado inicialmente para la pantalla de administración de proyectos y usuarios.



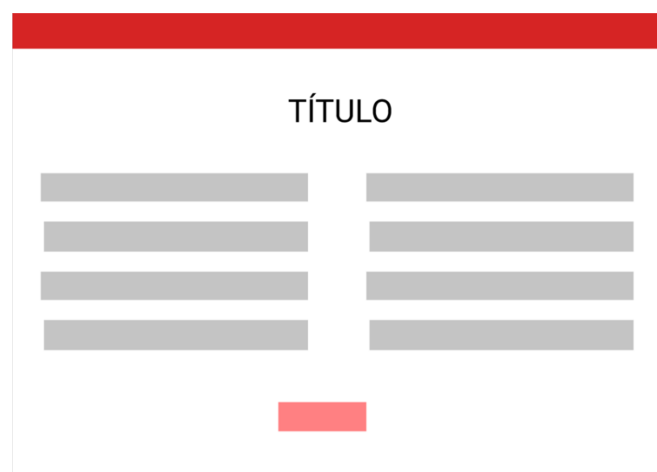
Anexo A.2: Prototipo de la interfaz para motor de búsqueda.

El Anexo A.2 muestra el prototipo presentado inicialmente para la pantalla de motor de búsqueda.



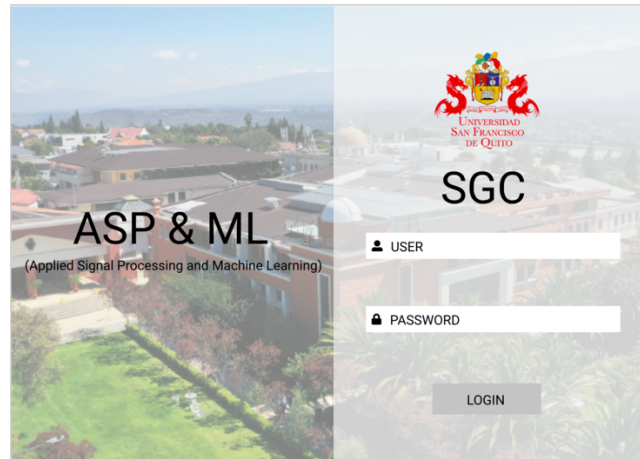
Anexo A.3: Prototipo de la interfaz de menú.

El Anexo A.3 muestra el prototipo presentado inicialmente para la pantalla principal del sistema una vez iniciado sesión dentro del mismo.



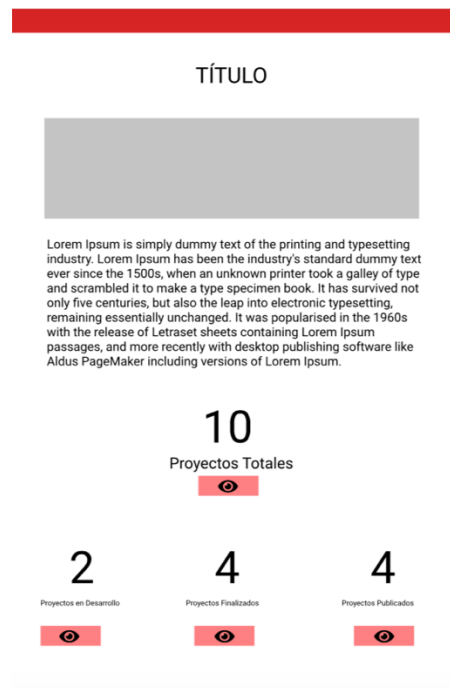
Anexo A.4: Prototipo de la interfaz para formularios.

El Anexo A.4 muestra el prototipo presentado inicialmente desde el cual se basan todos los formularios dentro del sistema.



Anexo A.5: Prototipo de la interfaz para página de inicio de sesión.

El Anexo A.5 muestra el prototipo presentado inicialmente para la pantalla de inicio de sesión del sistema.



Anexo A.6: Prototipo de la interfaz para pantalla inicial.

El Anexo A.6 muestra el prototipo presentado inicialmente para la pantalla inicial del sistema.