



UNIVERSIDAD SURCOLOMBIANA

GESTIÓN DE BIBLIOTECAS

CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

 iconia
ISO 9001

SA-CERE 59

The logo consists of a circular border containing the word "CERTIFIED" at the top and "MANAGEMENT SYSTEM" at the bottom, separated by a horizontal line. In the center, the word "IQNet" is written in a bold, sans-serif font.

Neiva, 29 de marzo de 2023

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

NEIVA

El (Los) suscrito(s):

Autor(es) de la tesis y/o trabajo de grado o

Titulado: Diseño e implementación de una plataforma para la administración de la producción intelectual de los docentes de la Universidad Surcolombiana

presentado y aprobado en el año 2023 como requisito para optar al título de

Ingeniero Electrónico

Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales “open access” y en las redes de información con las cuales tenga convenio la Institución.
 - Permite la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.
 - Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

Vigilada Mineducación

Vigilada Mineducación
La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



UNIVERSIDAD SURCOLOMBIANA
GESTIÓN DE BIBLIOTECAS

CARTA DE AUTORIZACIÓN



CÓDIGO	AP-BIB-FO-06	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 2
--------	--------------	---------	---	----------	------	--------	--------

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, "Los derechos morales sobre el trabajo son propiedad de los autores", los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

EL AUTOR/ESTUDIANTE:

Freddy Leonardo Cuenca Beltrán

Firma:

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



UNIVERSIDAD SURCOLOMBIANA
GESTIÓN DE BIBLIOTECAS

DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

CÓDIGO

AP-BIB-FO-07

VERSIÓN

1

VIGENCIA

2014

PÁGINA

1 de 4



TÍTULO COMPLETO DEL TRABAJO: DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA PARA LA ADMINISTRACIÓN DE LA PRODUCCIÓN INTELECTUAL DE LOS DOCENTES DE LA UNIVERSIDAD SURCOLOMBIANA

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
CUENCA BELTRÁN	FREDDY LEONARDO

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
CORTÉS CABEZAS	ALBEIRO
SALCEDO MORILLO	DIXON

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre

PARA OPTAR AL TÍTULO DE: INGENIERO ELECTRÓNICO

FACULTAD: INGENIERÍA

PROGRAMA O POSGRADO:

INGENIERÍA ELECTRÓNICA

CIUDAD: NEIVA AÑO DE PRESENTACIÓN: 2023 NÚMERO DE PÁGINAS: 50

TIPO DE ILUSTRACIONES (Marcar con una X):

Diagramas Fotografías Grabaciones en discos Ilustraciones en general Grabados Láminas
Litografías Mapas Música impresa Planos Retratos Sin ilustraciones Tablas o Cuadros

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

CÓDIGO

AP-BIB-FO-07

VERSIÓN

1

VIGENCIA

2014

PÁGINA

2 de 4



SOFTWARE requerido y/o especializado para la lectura del documento:

NO

MATERIAL ANEXO:

MANUAL DEL PROGRAMADOR

MANUAL DE USUARIO

PREMIO O DISTINCIÓN (*En caso de ser LAUREADAS o Meritoria*):

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

<u>Español</u>	<u>Inglés</u>	<u>Español</u>	<u>Inglés</u>
1. HTML5	HTML5	6. Plataforma	Website
2. Javascript	Javascript	7. Interfaz	Interface
3. JQuery	JQuery	8. Proceso	Process
4. NodeJS	NodeJS	9. Formulario	Form
5. MongoDB	MongoDB	10. Revisión	Audit

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

En este proyecto se expone el desarrollo e implementación de una plataforma web que permite la administración de la producción intelectual de los docentes de la Universidad Surcolombiana. Esta producción tiene un proceso de solicitud, revisión y evaluación por parte del Comité de Asignación de Puntaje. Por lo tanto, la directriz de la plataforma es facilitar, ordenar y adaptarse a las necesidades de los usuarios implicados en este proceso. La plataforma tiene como base las tecnologías HTML5, Javascript, JQuery, NodeJS, MongoDB entre otras. La finalidad de este aplicativo web es la generación de informes con toda la información obtenida, agrupada según las necesidades propuestas por la comunidad académica, así mismo, otorgar una interfaz de fácil entendimiento por parte de los usuarios como la adaptabilidad a los diferentes dispositivos electrónicos compatibles con la web.



UNIVERSIDAD SURCOLOMBIANA
GESTIÓN DE BIBLIOTECAS

DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

CÓDIGO

AP-BIB-FO-07

VERSIÓN

1

VIGENCIA

2014

PÁGINA

3 de 4



ABSTRACT: (Máximo 250 palabras)

This project presents the development and implementation of a web platform that allows the administration of the intellectual production of the teachers of the Universidad Surcolombiana. This production has a process of application, review and evaluation by the Score Assignment Committee. Therefore, the guideline of the platform is to facilitate, organize and adapt to the needs of the users involved in this process. The platform is based on HTML5, Javascript, JQuery, NodeJS, MongoDB and other technologies. The purpose of this web application is to generate reports with all the information obtained, grouped according to the needs proposed by the academic community, as well as to provide an easy-to-understand interface for users and adaptability to different electronic devices compatible with the web.

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.

	UNIVERSIDAD SURCOLOMBIANA GESTIÓN DE BIBLIOTECAS						
DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO							
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	4 de 4

APROBACION DE LA TESIS

Nombre Jurado: JOHAN JULIÁN MOLINA MOSQUERA

Firma:



Nombre Jurado: MARTÍN DIOMEDES BRAVO OBANDO

Firma:



Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.

DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA PARA LA
ADMINISTRACIÓN DE LA PRODUCCIÓN INTELECTUAL DE LOS DOCENTES
DE LA UNIVERSIDAD SURCOLOMBIANA

FREDDY LEONARDO CUENCA BELTRÁN

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA
2023

DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA PARA LA
ADMINISTRACIÓN DE LA PRODUCCIÓN INTELECTUAL DE LOS DOCENTES
DE LA UNIVERSIDAD SURCOLOMBIANA

FREDDY LEONARDO CUENCA BELTRÁN
2010295844

PROYECTO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO
ELECTRÓNICO

DIRECTOR:
ALBEIRO CORTÉS CABEZAS
Ingeniero Electrónico
Doctor en Ingeniería y Magister en Ingeniería Eléctrica y Electrónica

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA
2023

Nota de Aceptación

Director
Albeiro Cortés Cabezas

Evaluador 1
Martín Diomedes Bravo Obando

Evaluador 2
Johan Julián Molina Mosquera

DEDICATORIA

Este trabajo está dedicado a mis padres y hermanas quienes me han apoyado durante mi carrera universitaria.

AGRADECIMIENTOS

Agradezco inmensamente a mi familia, principal motor para lograr alcanzar este objetivo en mi vida. A mi director de proyecto de grado, el ingeniero pHD. Albeiro Cortés. A Liliana Santofimio por compartir toda su experiencia y mi pareja Valeria Hurtado por su apoyo incondicional.

CONTENIDO

	Pág.
DEDICATORIA.....	4
AGRADECIMIENTOS.....	5
LISTA DE FIGURAS.....	7
LISTA DE ANEXOS.....	9
RESUMEN.....	10
ABSTRACT.....	11
1. INTRODUCCIÓN.....	12
2. OBJETIVOS	13
2.1 OBJETIVO GENERAL.....	13
2.2 OBJETIVOS ESPECÍFICOS	13
3. PLANTEAMIENTO DEL PROBLEMA.....	14
3.1 FORMULACIÓN DEL PROBLEMA	14
3.2 JUSTIFICACIÓN	14
4. MARCO TEÓRICO	16
5. METODOLOGÍA.....	23
6. DESARROLLO DEL PROYECTO	35
7. ANÁLISIS DEL DESARROLLO DEL PROYECTO	42
8. CONCLUSIONES.....	43
9. RECOMENDACIONES.....	44
10. BIBLIOGRAFÍA.....	45

LISTA DE FIGURAS

	Pág
Figura 1. Ejemplo estandarizado de “Hola mundo” en Node.js.	20
Figura 2. Ejemplo estandarizado de la conexión de MongoDB con el módulo mongoose de Node.js.	22
Figura 3. Ejemplo estandarizado de “Hola mundo” de Express.js.	23
Figura 4. Modelo de schema de usuario.	23
Figura 5. Ejemplo de un documento para el usuario administrador.	24
Figura 6. Ejemplo de un documento para el usuario Funcionario.	24
Figura 7. Ejemplo de un documento para el usuario Docente.	25
Figura 8. Modelo schema de una solicitud.	26
Figura 9. Modelo schema de un autor de una solicitud.	27
Figura 10. Modelo schema de ISSN.	27
Figura 11. Modelo schema de ISBN.	28
Figura 12. Diagrama de flujo del proceso principal de la plataforma.	29
Figura 13. Diagrama de flujo del primer usuario admin de la plataforma.	30
Figura 14. Diagrama de flujo del inicio de sesión en la plataforma.	31
Figura 15. Diagrama de flujo del proceso para cargar ISSN o ISBN.	32
Figura 16. Diagrama de flujo de la creación de una solicitud de puntuación de productividad académica.	33
Figura 17. Diagrama de flujo de la generación de informe.	34

Figura 18. Formulario para agregar un usuario a la plataforma.	35
Figura 19. Listado de todos los usuarios.	36
Figura 20. Listado de seriales ISSN.	36
Figura 21. Listado de seriales ISBN.	37
Figura 22. Formulario para la creación de una solicitud.	38
Figura 23. Listado de solicitudes de un docente.	38
Figura 24. Listado de solicitudes pendientes por revisión.	39
Figura 25. Interfaz de la primera revisión.	40
Figura 26. Formulario de revisión CAP.	40
Figura 27. Interfaz de generación de informe.	41

LISTA DE ANEXOS

	Pág
Anexo 1. Formato MI-INV-FO-05.	47
Anexo 2. Carta respuesta de Vicerrectoría de Investigación y Proyección Social para solicitud de información para pruebas finales.	48
Anexo 3. Formato para cargar ISSN.	49
Anexo 4. Formato para cargar ISBN.	49
Anexo 5. Informe general.	50

RESUMEN

En este proyecto se expone el desarrollo e implementación de una plataforma web que permite la administración de la producción intelectual de los docentes de la Universidad Surcolombiana. Esta producción tiene un proceso de solicitud, revisión y evaluación por parte del Comité de Asignación de Puntaje. Por lo tanto, la directriz de la plataforma es facilitar, ordenar y adaptarse a las necesidades de los usuarios implicados en este proceso. La plataforma tiene como base las tecnologías HTML5, Javascript, JQuery, NodeJS, MongoDB entre otras. La finalidad de este aplicativo web es la generación de informes con toda la información obtenida, agrupada según las necesidades propuestas por la comunidad académica, así mismo, otorgar una interfaz de fácil entendimiento por parte de los usuarios como la adaptabilidad a los diferentes dispositivos electrónicos compatibles con la web.

PALABRAS CLAVE: HTML5, Javascript, JQuery, NodeJS, MongoDB, Plataforma, Interfaz, Proceso.

ABSTRACT

This project presents the development and implementation of a web platform that allows the administration of the intellectual production of the teachers of the Universidad Surcolombiana. This production has a process of application, review and evaluation by the Score Assignment Committee. Therefore, the guideline of the platform is to facilitate, organize and adapt to the needs of the users involved in this process. The platform is based on HTML5, Javascript, JQuery, NodeJS, MongoDB and other technologies. The purpose of this web application is to generate reports with all the information obtained, grouped according to the needs proposed by the academic community, as well as to provide an easy-to-understand interface for users and adaptability to different electronic devices compatible with the web.

KEYWORDS: HTML5, Javascript, JQuery, NodeJS, MongoDB, Platform, Interface, Process.

1. INTRODUCCIÓN

En Colombia se suelen utilizar plataformas web para la gestión general de las IES. Su función principal es la administración, organización y búsqueda del material intelectual tanto de docentes como estudiantes, así como el almacenamiento de resultados de proyectos e investigaciones. Sin embargo, la capacidad de dar estadísticas, recuentos y registros es inexistente, ya que estas herramientas no van más allá de su función básica de búsqueda y almacenamiento. Algunas Instituciones de Educación Superior (IES) colombianas realizan el archivo de la producción intelectual de sus docentes (investigaciones, artículos y demás material relacionado) en carpetas físicas con documentos impresos, con limitantes de espacio, uso y optimización de los conocimientos obtenidos (RAGGETT, David *et al.*). Esta situación puede considerarse obsoleta y poco eficiente teniendo en cuenta las diversas herramientas tecnológicas que pueden ser utilizadas para lograr una administración productiva del conocimiento.

La Universidad Surcolombiana ha venido desarrollando en su plataforma web un buscador gestionado por la biblioteca universitaria. En su proceso de apoyo No.18 “Gestión de Servicios Bibliotecarios”, que hace parte del “Sistema de Gestión de Calidad” (SGC), se ha puesto como objetivo proveer a los actores interesados, información cultural y académica de forma física y virtual a través de la documentación almacenada en sus bases de datos (BIBEAULT, Bear; DE ROSA, Aurelio y KATZ, Yehuda). Echando un vistazo al abanico de servicios que provee la plataforma bibliotecaria de la USCO, este se cierra a la función básica de consulta sobre referencias, periódicos, trabajos de tesis, revistas y bases de datos.

Por otro lado, carece de un sistema adecuado de explotación de datos para transformarlos en estadísticas que proporcionen transparencia y comprensibilidad al usuario, y que apoyen la toma de decisiones dentro del ambiente administrativo y educativo de la institución. La plataforma web gestionada por la biblioteca de la USCO, cumple con el requisito básico de búsqueda y almacenamiento, pero deja de lado la gestión de estadísticas y el recuento de la productividad de la Universidad en materia intelectual. Cabe aclarar que una herramienta con la capacidad de dar manejo, recuento y estadísticas del material intelectual, proporcionaría una base para la asignación de puntajes dados por el CAP, el cual puede presentar errores como la contabilización doble de la productividad académica no permitida en el artículo 10 del Decreto Ley 1279.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Diseñar e implementar una plataforma web para la administración de la producción intelectual de los docentes en la Universidad Surcolombiana.

2.2 OBJETIVOS ESPECÍFICOS

- Implementar una plataforma web capaz de proveer estadísticas, dar seguimiento y realizar informes periódicos de la producción intelectual de los docentes.
- Diseñar una interfaz gráfica amigable, clara y ordenada.
- Validar la plataforma mediante pruebas piloto con usuarios finales.

3. PLANTEAMIENTO DEL PROBLEMA

De acuerdo a Ortiz (2016), actualmente el sector de la educación superior en Colombia no ha identificado ni potenciado de forma intensiva su capital estructural que incluye softwares informáticos, documentos digitales y de hardware (CEDERHOLM, Dan y ZELDMANN, Jeffrey). Este sistema de archivo es paquidérmico cuando los usuarios de esta información requieren realizar consultas, obtener estadísticas, hacer seguimiento o aprovechar los avances e investigaciones realizadas al presentarse gran aglomeración del material que no ha sido clasificado con efectividad.

Dentro de las consideraciones anteriores, La Universidad Surcolombiana (USCO) no es una excepción. En la actualidad, la comunidad educativa expresa constantemente la necesidad de un ambiente informático óptimo de gestión para la administración de investigaciones y artículos correspondientes a la producción intelectual de cada docente. Por ejemplo, la misma comunidad académica resalta que el trámite administrativo y de información para la asignación de puntos de acuerdo a la productividad académica de cada docente, que se expresa en el Decreto Ley 1279 y asignados por el Comité de Asignación de Puntaje (CAP), presenta fallas protuberantes en su gestión (GASSTON, Peter).

3.1 FORMULACIÓN DEL PROBLEMA

Lo anterior sugiere que no existe una herramienta capaz de proporcionar un informe detallado de la gestión/administración de la producción académica universitaria. En ese sentido, nos preguntamos: ¿La tecnología web puede proporcionar una ayuda significativa para el manejo de la producción intelectual de la USCO?; si es así, ¿de qué manera se puede diseñar e implementar un modelo acorde a las necesidades de la comunidad académica?

3.2 JUSTIFICACIÓN

Las bases de datos han llegado junto con nuevas herramientas informáticas para facilitar el manejo de grandes cantidades de información. Para dar solución a la aglomeración de documentos y su administración ineficiente es recomendable implementar una plataforma capaz de administrar la producción intelectual desarrollada por los docentes en IES, facilitando su consulta, seguimiento, investigación y clasificación (CORRADI, Valentina y SWANSON, R Norman).

De acuerdo a Chua (2004), un sistema de administración del conocimiento apoyado en un aplicativo web es importante porque impulsa la productividad y la habilidad de toma de decisiones de una organización, en este orden de ideas, se construye un sistema de trabajo inteligente capaz de producir información rápida, sistematizada, entendible e informada (BODIN, Martin *et al*).

Para el caso particular de la USCO, la implementación de un aplicativo web para la gestión de la información mejora la aplicación del Decreto Ley 1279 de 2012, con el cual, se estipula la bonificación salarial a los docentes por los resultados derivados de investigaciones y actividades académicas y que es asignado por parte del Comité de Asignación de Puntaje (CAP). Los informes generados por esta plataforma darán soporte para la asignación de esta puntuación y apoyarán la toma de decisiones (SEONGHOON, Kang y SUKYOUNG, Ryu).

4. MARCO TEÓRICO

4.1 HTML 5

HTML es un lenguaje que permite definir la estructura y contenido de una página web o un documento. En un principio, este lenguaje fue utilizado exclusivamente para crear documentos electrónicos, sin embargo, en la actualidad se utiliza en muchas aplicaciones electrónicas como banca electrónica, buscadores, tiendas online, entre otros (RAGGETT, David; LE HORS, Arnaud y JACOBS, Ian).

HTML5 es la quinta versión del lenguaje de programación HTML para describir el contenido y la apariencia de las páginas web. Se desarrolló para abordar los problemas de compatibilidad con el estándar actual, HTML4. Esta versión, a diferencia de versiones anteriores, no requiere el uso de API o complementos propietarios. HTML5 ofrece una interfaz común que facilita la carga de elementos y hace innecesaria la instalación de un complemento Flash, por ejemplo, ya que estos elementos se ejecutan solos.

Uno de los objetivos detrás de la creación de HTML5 es la compatibilidad con documentos multimedia en dispositivos móviles. Para ello se han creado funciones sintácticas como las etiquetas de video, audio y canvas. Las nuevas características de HTML5 realmente cambian la forma en que interactúa con los documentos. Por ejemplo: nuevas reglas de análisis para mayor flexibilidad, nuevos atributos, eliminar atributos obsoletos o redundantes, la capacidad de arrastrar y soltar elementos de un documento HTML5 a otro, edición fuera de línea, funciones de mensajería mejoradas, reglas detalladas para el análisis, registrar manejadores de protocolos y MIME, un estándar común para almacenar datos en bases de datos SQL (Web SQL).

HTML5 fue adoptado en 2007 por el nuevo grupo de trabajo del World Wide Web Consortium (W3C). Este grupo publicó la primera versión de la norma en enero de 2008.

4.2 CSS 3

CSS (Cascading Style Sheets) es un lenguaje que ayuda a crear páginas web de apariencia ordenada. Los documentos en cuestión son archivos de texto estructurados con un lenguaje de marcas: HTML es el más conocido de estos

lenguajes, otros ejemplos son SVG o XML (CEDERHOLM, Dan y ZELDMANN, Jeffrey).

Presentar un documento al usuario significa convertir este documento en un formato utilizable por la audiencia prevista. Los navegadores, como Firefox, Chrome, Safari o Edge están diseñados para presentar documentos visualmente, ya sea en la pantalla de una computadora, un proyector de video o una impresora.

CSS se puede usar para formatear documentos básicos, por ejemplo, cambiar el color y el tamaño de los títulos y enlaces. Se puede usar para hacer un diseño, por ejemplo, transformar el texto que se muestra en una columna en una composición con un marco principal y una barra lateral para información relacionada. Con CSS, también se puede producir animaciones (GASSTON, Peter).

Cuando HTML se propaga por el boom de internet, la W3C decidió estandarizar CSS y en 1996 se publicó en el primer estándar o CSS 1. Con el tiempo aparecieron nuevos estándares como CSS 2, CSS 2.1 y finalmente CSS3.

CSS3 se ha dividido en "módulos". Contiene la "especificación CSS antigua" (que se ha dividido en partes más pequeñas). Además, se añaden nuevos módulos. Algunos de los módulos CSS3 más importantes son: Selectors, Box Model, Backgrounds and Borders, Image Values and Replaced Content, Text Effects, 2D/3D Transformations, Animations, Multiple Column Layout, User Interface.

La mayoría de las nuevas propiedades de CSS3 se implementan en navegadores modernos.

4.3 **Bootstrap**

Bootstrap es un framework HTML, CSS y JavaScript gratuito y de código abierto para crear rápidamente sitios web responsive. Inicialmente, Bootstrap se llamaba Twitter Blueprint y fue desarrollado por un equipo que trabajaba en Twitter. Es compatible con el diseño responsive y ofrece plantillas de diseño prefabricadas que puede usar directamente o personalizar según sea necesario con su código. Es muy útil por la compatibilidad entre navegadores. Bootstrap es compatible con todos los navegadores modernos y las versiones más recientes de Internet Explorer (CORRADI, Valentina y SWANSON, R Norman).

Una de sus ventajas principales se evidencia en la facilidad para implementar el diseño responsive, pudiendo mostrar la información del sitio web en diferentes

dispositivos electrónicos independientemente del tamaño del mismo. Utiliza un sistema de columnas, y divide la pantalla en doce de estas, donde se puede indicar el tamaño de columna de acuerdo con el tamaño de pantalla del dispositivo.

Bootstrap otorga uniformidad a la página y una apariencia moderna. Algunos de los plugins de Javascript, que son componentes basados en la librería jQuery permiten esta funcionalidad y en los cuales podemos encontrar Carousel, Collapse Modal, Tooltip, Dropdown, Scrollspy, Tab, Popover, Alert, y Typehead.

4.4 Java Script

JavaScript es un lenguaje de secuencias de comandos multiplataforma, liviano y orientado a objetos, diseñado principalmente para agregar interactividad a las páginas y aplicaciones web modernas. Es uno de los tres componentes principales de las tecnologías front-end además de HTML (que proporciona estructura a la página web) y CSS (que proporciona estilo a la página web). Fue desarrollado por un programador estadounidense Brendan Eich en Netscape en 1995. Originalmente llamado por los nombres 'Mocha' y 'LiveScript', Javascript obtuvo su nombre actual en diciembre de 1995 con el lanzamiento de la tercera versión beta (BODIN, Martin *et al*).

JavaScript se envió a ECMA (Asociación Europea de Fabricantes de Computadoras) para su estandarización, de modo que otros navegadores pudieran implementarlo sobre la base de la estandarización. ECMA International luego publicó las especificaciones estándar del lenguaje JavaScript bajo el nombre ECMAScript. La especificación de JavaScript actual se basa en ECMAScript 2016, que se lanzó en junio de 2016.

JavaScript proporciona funciones ricas para crear interfaces dinámicas e interactivas en aplicaciones del lado del cliente. Puede detectar el navegador y el sistema operativo del usuario (Sistema operativo), y realiza operaciones específicas de la plataforma. Originalmente denominado como un lenguaje interpretado, JavaScript se puede ejecutar sin una compilación preliminar por parte del navegador para que pueda realizar cálculos simples en el lado del cliente. A menudo se usa para validar la entrada del usuario en los formularios y enviar los datos de forma asíncrona con AJAX. También realiza la manipulación HTML DOM para proporcionar una interfaz dinámica. Es un lenguaje de programación orientado a objetos y admite varios objetos integrados con herencia. Por lo tanto, JavaScript permite a los desarrolladores agregar funciones dinámicas a páginas HTML.

estáticas, controlar multimedia y agregar animaciones. Además, JavaScript proporciona varias API y bibliotecas de terceros para facilitar la tarea de crear páginas web dinámicas. Una de las bibliotecas de JavaScript más populares es JQuery (SEONGHOON, Kang y SUKYOUNG, Ryu).

4.5 **JQuery**

Query , o jQuery , es una biblioteca de JavaScript gratuita, de código abierto y multiplataforma . Compatible con todos los navegadores web (Internet Explorer, Safari, Chrome, Firefox, etc.), fue diseñado y desarrollado en 2006 para facilitar la creación de scripts. Es el marco de JavaScript más conocido y más utilizado. Le permite actuar sobre códigos HTML, CSS, JavaScript y AJAX y, esencialmente, se ejecuta en el lado del cliente (BIBEAULT, Bear; DE ROSA, Aurelio y KATZ, Yehuda).

La librería jQuery permite, entre otras cosas, ganar velocidad en la interacción con el código HTML de una página Web. Ofrece como características principales: manipulación del modelo de objeto de documento (DOM), gestión de eventos (movimientos del ratón, clics, etc.) y AJAX (arquitectura informática), creación de efectos de animación, manipulación de hojas de estilo en cascada y muchos otros.

La biblioteca jQuery es una de las bibliotecas de JavaScript más utilizadas en el mundo hoy en día para desarrollar sitios web. No requiere instalación especial y se puede descargar directamente desde el sitio web oficial de la biblioteca. En teoría, un conocimiento básico de CSS, HTML y JavaScript y un editor de código simple o cualquier software FTP es suficiente para permitir su uso. El jQuery permite no tener que preocuparse por las diferentes versiones de JavaScript y, por lo tanto, limitar las pruebas impuestas al programador por el uso de JavaScript (KRÓL, Karol y SZOMOROVA, Lenka).

4.6 **Node.js**

NodeJS es una herramienta gratuita codificada en Javascript y orientada a aplicaciones de red. Esta herramienta de JavaScript se ha hecho famosa en el mundo del desarrollo web desde hace unos años. Además, es muy popular entre gigantes web como Netflix, PayPal, LinkedIn, Uber, NASA, etc. Es una plataforma basada en el motor JavaScript V8 de Chrome para desarrollar rápidamente aplicaciones de red rápidas y escalables. Como JavaScript es un lenguaje basado en eventos, Node.js en sí mismo está basado en eventos. Lo que le permite ser ligero y tener un rendimiento de muy alto nivel. Node.js no es ni un servidor ni un

framework. Es solo un entorno de tiempo de ejecución de JavaScript que se puede usar para escribir servicios del lado del servidor llamados API (interfaz de programación de aplicaciones). Es una alternativa a lenguajes de servidor como PHP, Java EE, etc (CANTELON, Mike *et al*).

Node.JS también es adoptado por los desarrolladores web en estos días debido a que ofrece ventajas muy interesantes. Node.js es ideal para RTA y SPA, estas son aplicaciones que realizan muchas solicitudes de archivos. Y dado que Node.js está equipado con un modelo sin bloqueo, puede realizar varias acciones simultáneamente y no una tras otra como sería el caso con un modelo de bloqueo (TEIXEIRA, Pedro).

Node.js es una plataforma muy flexible. Con muy pocas opciones ya integradas, dejando al usuario la opción de integrar los módulos que contendrán las funciones que desee. Además, le proporciona bibliotecas de funciones. Estos le permiten realizar acciones como escribir en la salida estándar, abrir/cerrar conexiones de red o incluso crear un archivo.

Otra ventaja de Node.js es su uso con JavaScript. Es un lenguaje que no es nada difícil de aprender y es conocido por todos. En la figura 1 se observa la versión de un “Hola mundo” de un servidor HTTP escrito en Note.js.

```
1 const http = require('http');
2
3 const hostname = '127.0.0.1';
4 const port = 3000;
5
6 const server = http.createServer((req,res) => {
7     res.statusCode = 200;
8     res.setHeader('Content-Type', 'text/plain');
9     res.end('Hola Mundo\n');
10 });
11
12 server.listen(port, hostname, () => {
13     console.log(`El servidor se está ejecutando en http://\${hostname}:\${port}/`\)
```

Figura 1. Ejemplo estandarizado de “Hola mundo” en Node.js.

4.7 MongoDB

MongoDB es una base de datos NoSQL orientada a documentos que se utiliza para el almacenamiento de datos de gran volumen. En lugar de usar tablas y filas como en las bases de datos relacionales tradicionales, MongoDB hace uso de colecciones

y documentos. Los documentos consisten en pares clave-valor que son la unidad básica de datos en MongoDB. Las colecciones contienen conjuntos de documentos y funciones que son equivalentes a las tablas de bases de datos relacionales. MongoDB es una base de datos que salió a la luz a mediados de la década de 2000. (EYADA, Mahmoud M et al).

Algunas características de Mongo DB son las siguientes:

- Cada base de datos contiene colecciones que a su vez contienen documentos. Cada documento puede ser diferente con un número variable de campos. El tamaño y el contenido de cada documento pueden ser diferentes entre sí.
- La estructura del documento está más en línea con la forma en que los desarrolladores construyen sus clases y objetos en sus respectivos lenguajes de programación. Los desarrolladores suelen decir que sus clases no son filas y columnas, sino que tienen una estructura clara con pares clave-valor.
- Las filas (o documentos como se llama en MongoDB) no necesitan tener un esquema definido de antemano. En su lugar, los campos se pueden crear sobre la marcha.
- El modelo de datos disponible dentro de MongoDB le permite representar relaciones jerárquicas, almacenar matrices y otras estructuras más complejas con mayor facilidad.
- Los entornos de MongoDB son muy escalables. Empresas de todo el mundo han definido clústeres y algunos de ellos ejecutan más de 100 nodos con alrededor de millones de documentos dentro de la base de datos.

En la figura 2 es un ejemplo estandarizado de la conexión de MongoDB con el módulo mongoose de Node.js.

```

const mongoose = require('mongoose')

let PID_APP_HOST='127.0.0.1'
let PID_APP_DATABASE='PID_DB'

const MONGODB_URI = `mongodb://${PID_APP_HOST}/${PID_APP_DATABASE}`;

mongoose.connect(MONGODB_URI, {
  useNewUrlParser: true,
  useUnifiedTopology: true
})
.then(db => console.log(`Database is connected`))
.catch(err => console.log(err));

```

Figura 2. Ejemplo estandarizado de la conexión de MongoDB con el módulo mongoose de Node.js.

4.8 Express.js

Express es un framework, escrito en JavaScript y alojado en el entorno de tiempo de ejecución de Node.js. Es el marco web de nodo más popular y es la biblioteca subyacente para una serie de otros marcos web de nodo populares (HAHN, Evan M). Proporciona mecanismos para:

- Escribir controladores para solicitudes con diferentes verbos HTTP a diferentes rutas URL (rutas).
- Integrar motores de renderizado "vue" para generar respuestas mediante la inserción de datos en los modelos.
- Definir la configuración común de la aplicación web, como el puerto que se utilizará para la conexión y la ubicación de las plantillas utilizadas para representar la respuesta.
- Agregar middleware de procesamiento de solicitudes adicional en cualquier momento en la canalización de procesamiento de solicitudes.

En la figura 3 se observa la versión de “Hola mundo” de un servidor de HTTP escrito en Express.js

```

1  const express = require('express')
2  const app = express()
3  const port = 3000
4
5  app.get('/', (req, res) => {
6    | res.send('Hello World!')
7  })
8
9  app.listen(port, () => {
10   | console.log(`Example app listening on port ${port}`)
11 })

```

Figura 3. Ejemplo estandarizado de “Hola mundo” de Express.js.

5. METODOLOGÍA

El presente proyecto consistió en el desarrollo de una plataforma web que consistió en la realización de alrededor de 4500 líneas de código, y debido a su extensión se dividió en procesos para su explicación.

5.1. MODELOS DE LA BASE DE DATOS

5.1.1. COLECCIONES

5.1.1.1. MODELO DEL USUARIO (User.js)

En la figura 4 se observa el modelo que especifica los datos requeridos para los tres roles de usuario: Admin, Funcionario y Docente. Los tres comparten el mismo schema diferenciado por el valor en la propiedad role.

```

const UserSchema = Schema({
  identification_type: {type: String},
  identification: { type: String, required: true, unique: true},
  name: {type: String, required: true},
  lastname: {type: String, required: true},
  sec_lastname: {type: String},
  role: {type: String, required: true, enum: ["Admin", "Funcionario", "Docente"]},
  email: {type: String, required: true, unique: true},
  password: {type: String, required: true},
  facultad: {type: String},
  programa: {type: String},
  vinculacion: {type: Boolean},
  investigacion: {type: Boolean},
  nombreGrupo: {type: String},
  lineaInvestigacion: {type: String},
  last_login_date: {type: Date, default: Date.now()}
}, {timestamps: true});

```

Figura 4. Modelo de schema de usuario.

Las propiedades principales de este modelo son las que se encuentran requeridas (*required: true*), siendo el número de identificación, el nombre completo, role, usuario y contraseña, estos dos últimos son los datos validados en el momento del inicio de la sesión mediante el módulo Passport de npm. Un ejemplo de un documento para el usuario administrador en la colección users es la figura 5.

```
{
  "_id": {
    "$oid": "63b6edde9b6ade4e79de47f2"
  },
  "identification_type": "Cedula",
  "identification": "12345678",
  "name": "administrador",
  "lastname": "admin",
  "sec_lastname": "admin",
  "role": "Admin",
  "email": "ejemploAdmin@ejemplo.com",
  "password": "$2a$10$H2xKGW7Va1sLGzyOxjYGL.154o597PXDk6nElkoBZwJlOujR9uFlw",
  "last_login_date": {
    "$date": "2023-01-09T14:51:19.777Z"
  },
  "createdAt": {
    "$date": "2023-01-05T15:33:50.376Z"
  },
  "updatedAt": {
    "$date": "2023-01-09T14:51:19.778Z"
  },
  "__v": 0
}
```

Figura 5. Ejemplo de un documento para el usuario administrador.

Las variables no requeridas facultad, programa, vinculacion, investigacion, nombreGrupo, lineaInvestigacion serán *null* para este usuario Admin, así como para el Funcionario en la figura 6.

```
{
  "_id": {
    "$oid": "63b720b39b6ade4e79de487c"
  },
  "identification_type": "Cedula",
  "identification": "567898765",
  "name": "nombre_funcionario",
  "lastname": "apellido_funcionario",
  "sec_lastname": "",
  "role": "Funcionario",
  "email": "ejemploFuncionario@ejemplo.com",
  "password": "$2a$10$mdHsHc0dIKHtREH5n18B6ONjx4/mgnpNKeQ4YYY7WhzzjDjPeMyvu",
  "last_login_date": {
    "$date": "2023-01-05T23:40:55.040Z"
  },
  "createdAt": {
    "$date": "2023-01-05T19:10:43.227Z"
  },
  "updatedAt": {
    "$date": "2023-01-05T23:40:55.040Z"
  },
  "__v": 0
}
```

Figura 6. Ejemplo de un documento para el usuario Funcionario.

Los docentes tienen la propiedad facultad y programa para futuros filtros. Las propiedades de vinculacion, investigacion, nombreGrupo, lineaInvestigacion, son requerimientos del formato MI-INV-FO 05. Un ejemplo de un documento para el usuario Docente en la colección users es la figura 7.

```
{  
    "_id": {  
        "$oid": "63b6f60b9b6ade4e79de4824"  
    },  
    "identification_type": "Cedula",  
    "identification": "123456789",  
    "name": "nombre_Docente",  
    "lastname": "apellido_Docente",  
    "sec_lastname": "segundo_apellido_Docente",  
    "role": "Docente",  
    "email": "ejemploDocente@ejemplo.com",  
    "password": "$2a$10$WCYxsYIqBN.qeBupqw1GDOAJ1DMRJBaq6dt7piLDfRMx4Wa3lW",  
    "facultad": "Ingeniería",  
    "programa": "Ingeniería Agrícola",  
    "vinculacion": false,  
    "investigacion": false,  
    "last_login_date": {  
        "$date": "2023-01-05T23:19:40.776Z"  
    },  
    "createdAt": {  
        "$date": "2023-01-05T16:08:43.581Z"  
    },  
    "updatedAt": {  
        "$date": "2023-01-05T23:19:40.776Z"  
    },  
    "__v": 0  
}
```

Figura 7. Ejemplo de un documento para el usuario Docente.

5.1.1.2. MODELO DE SOLICITUDES (publications.js)

Esta colección guarda las solicitudes creadas por los docentes (Figura 8) y el estado en el que se encuentra la solicitud que consta de seis estados:

1. “Pendiente por revisión”: En este estado la solicitud está recién creada por el docente y puesta en lista de pendiente por revisión del funcionario.
2. “Revisado”: El funcionario ha aceptado la solicitud para presentarla ante el CAP.
3. “Editar”: El funcionario ha decidido devolver la solicitud con la opción de modificar la solicitud junto a una observación especificando las correcciones a efectuar.
4. “Rechazado”: Cuando una solicitud se encuentra con datos erróneos sin opción de corregir, el funcionario podrá rechazar totalmente esta solicitud y el docente deberá crear una nueva.
5. “No aprobado por CAP”: Una solicitud validada por el funcionario se someterá ante el CAP. Si este comité decide no otorgar puntos por productividad

- académica la solicitud se encontrará en estado “No aprobado por CAP”, junto al concepto emitido por él.
6. “Aprobado”: En caso de que el CAP otorgue puntos a la productividad académica, el estado de esta solicitud será “Aprobado”, podrá dar valor a la propiedad “puntaje”, qué tipo de puntaje (*true* = Bonificación por productividad académica o *false* = Reconocimiento de puntos salariales) y el concepto por el cual se aprobó.

```
const PublicationSchema = Schema({
  id_Docente: Schema.Types.ObjectId,
  name: {type: String, required: true},
  modalidad: {type: String, required: true},
  categoria: {type: String},
  tipo: {type: String},
  nombre_revista:{type: String},
  tiempo_revista:{type: Number},
  fecha_recepcion_revista:{type: Date},
  fecha_publicacion: {type: Date},
  ISBN:{type:String}, //ISSN ó ISBN
  recursos_U:{type:Boolean},
  nombre_proyecto_investigacion:{type: String},
  editorial:{type:String},
  URL: {type: String},
  cambio_categoria: {type: Boolean},
  //Datos del Archivo
  filename: {type: String},
  path: {type: String},
  originalname: {type: String},
  mimetype: {type: String},
  size: {type: Number},
  //Revisores
  estado: {type: String,enum: ['Pendiente por revisión','Revisado', 'Editar',
  'Rechazado','No aprobado por CAP','Aprobado'],
  default: 'Pendiente por revisión'},
  observacion:{type: String},
  conceptoCAP:{type: String},
  tipo_puntaje:{type: Boolean},
  puntaje: {type: Number}
},{timestamps: true});
```

Figura 8. Modelo schema de una solicitud.

La propiedad “ISBN” almacena el valor del ISSN o ISBN para las diferentes modalidades que lo requieran, siendo *null* para las que no dependan de una revista indexada o libro.

Las propiedades “modalidad”, “categoría” y “tipo”, varían según la clasificación que tenga la modalidad, por ejemplo, la modalidad “Artículo de Revista” posee tipo de investigación, comunicación corta, revisión, reflexión o reporte de caso, y la revista en las categorías de A1, A2, B y C; en cambio una “Producción de videos, cinematográficas o fonográficas” tiene impacto internacional o nacional siendo este dato almacenado en tipo y la finalidad de la producción didáctico o documental almacenado como categoría. Quienes no posean tipo o categoría, su valor será *null*. Las propiedades que corresponden a los datos del archivo (filename, path, originalname, mimetype, size) tienen como finalidad el almacenamiento de los archivos cargados por los docentes en la solicitud.

5.1.1.3. MODELO DE AUTOR (autor.js)

Una productividad académica puede tener múltiples autores, por lo tanto, es una buena solución poseer una colección que brinde la opción de ser dinámica esta variable en una solicitud. La figura 9 muestra que las propiedades del modelo son “id_publication” que será la propiedad que almacena el indicador único de la solicitud con la cual está relacionada y el nombre del autor.

```
const Autor = Schema({
  id_publication: Schema.Types.ObjectId,
  nombre: {
    type: String,
    required: true
  },
  {
    timestamps: true
});
```

Figura 9. Modelo schema de un autor de una solicitud.

5.1.1.4. MODELO DE INTERNATIONAL STANDARD SERIAL NUMBER – NÚMERO INTERNACIONAL NORMALIZADO DE PUBLICACIONES SERIADAS (ISSN) (ISSN.js)

La plataforma permite cargar masivamente los datos de las revistas indexadas mediante un archivo de formato .xlsx con las características requeridas para dar una verificación de los números proporcionados por los docentes en las solicitudes que contengan estos seriales.

Actualmente un ISSN puede ser impreso, electrónico o tipo L. Si se desea cargar un serial de años anteriores que no posean estas descripciones, se almacenará este número en la propiedad “issn_impreso” por defecto. (Figura 10).

```
const ISSN = Schema({
  titulo: {type: String, required: true},
  issn_impreso:{type: String},
  issn_electronico:{type: String},
  issn_L:{type: String},
  institucion_editora:{type: String},
  categoria:{type:String, required: true},
  vigencia:{type: Number, required: true}
},{
  timestamps: true
});
```

Figura 10. Modelo schema de ISSN.

5.1.1.5. MODELO DE INTERNATIONAL STANDARD BOOK NUMBER - CÓDIGO INTERNACIONAL NORMALIZADO PARA LIBROS (ISBN) (ISBN.js)

Las modalidades de productividad académica “Libro” y “Capítulo del libro”, tienen como propiedad el número serial correspondiente al libro de la productividad. Las propiedades que definen este serial son título, si es un obra impresa o digital, lugar de procedencia y el numero serial del editor visualizados en la figura 11. Esta colección permite almacenar masivamente los ISBN para poder validar en la interfaz del funcionario cada serial.

```
const ISBN = Schema({
    ISBN:{type: String, required: true},
    titulo:{type: String, required: true},
    impreso:{type: Boolean},
    digital:{type: Boolean},
    procedencia:{type: String},
    editor:{type:String},
    yyyy_publicacion:{type:Number}
},{
    timestamps: true
});
```

Figura 11. Modelo schema de ISBN.

5.2. PROCESOS DE LA PLATAFORMA

5.2.1. PROCESO GENERAL

En la figura 12 se observa un resumen del proceso de la plataforma el cual consiste primeramente en la creación de la solicitud de productividad académica por parte del docente con los datos correspondientes a esta. Posteriormente, es revisada por el funcionario encargado de la plataforma haciendo una validación manual teniendo acceso a toda la información basada en el formulario MI-INV-FO-05 de la USCO (Anexo 1), dictaminando así un rechazo, corrección o aceptación de estos datos, junto a una observación si es necesaria. Esta decisión está basada en las experiencias y opiniones brindadas en las pruebas con los usuarios finales de la plataforma.

El rechazo es motivo de negación total respecto a la solicitud del docente, en cambio una corrección habilita la edición de esta en el listado de solicitudes de la interfaz del docente enviando una notificación al correo.

Al poseer una solicitud válida para la presentación ante el Comité de Asignación y Reconocimiento de Puntaje (CAP) esta se someterá a la revisión por parte de este

comité que decide la negación o reconocimiento del puntaje y el tipo de bonificación sí es otorgada. Este proceso finaliza con un correo enviado al docente con la decisión tomada por el comité.

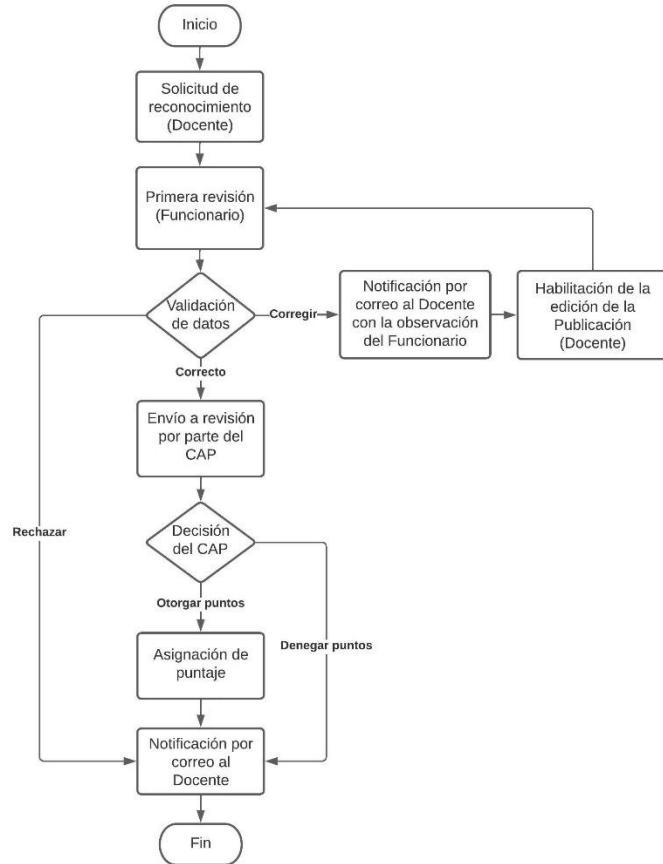


Figura 12. Diagrama de flujo del proceso principal de la plataforma.

5.2.2. PRIMER USUARIO ADMINISTRADOR DE LA PLATAFORMA

La base de datos no posee un administrador, por lo tanto, la ruta "<https://<dominio>/users/createAdminForm>" permite la creación del primer usuario Admin. A esta no se podrá acceder luego de la creación del primer administrador de la plataforma por lo que si existe se redirecciona a la vista de login. Este proceso se observa en la figura 13.

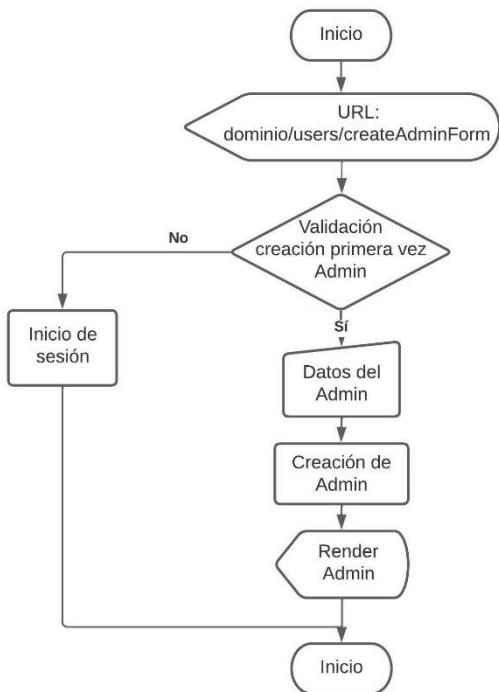


Figura 13. Diagrama de flujo del primer usuario admin de la plataforma.

5.2.3. INICIO DE SESIÓN EN LA PLATAFORMA

Para el uso del software, un usuario, previamente creado por el administrador de la plataforma, debe validar sus credenciales. La plataforma crea un registro de la última sesión abierta por este en su base de datos y con base en el rol preestablecido, redirecciona al usuario a su render predefinido. Dicho proceso se evidencia en la figura 14.

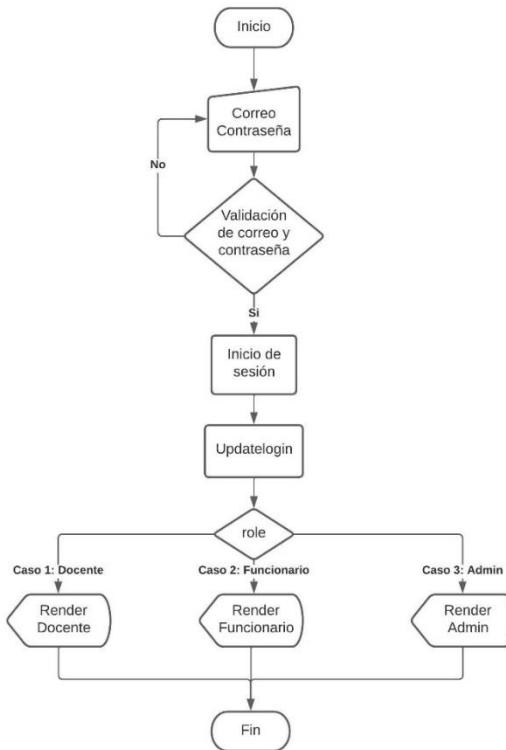


Figura 14. Diagrama de flujo del inicio de sesión en la plataforma.

5.2.4. PROCESO PARA CARGAR ISSN E ISBN

Las validaciones de los seriales para revistas indexadas o libros en la interfaz del funcionario se basan en las colecciones de ISSN e ISBN detallados en la sección de modelos esquemáticos.

Para cargar los índices seriados deben estar ordenados en el archivo .xlsx del anexo 3 para las revistas indexadas y anexo 4 para los libros publicados, indicando la información respectiva al encabezado de cada columna.

Los formatos con los datos se cargarán en la interfaz del administrador en el apartado correspondiente tanto para el ISSN como para el ISBN, diferenciándose así por el año de vigencia, la plataforma validará si el formato es correcto y la duplicidad que pueda haber con la información previamente almacenada. Este proceso esta descrito en la figura 15, donde finaliza con el almacenamiento de la información valida.

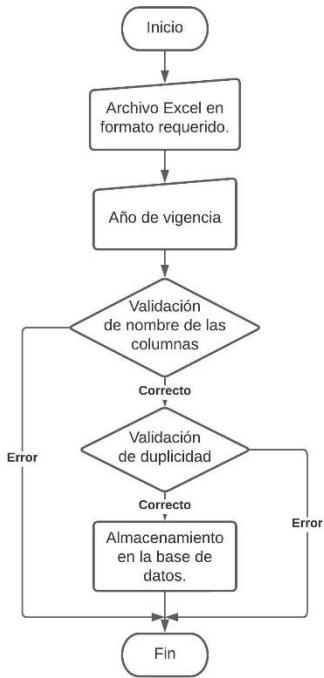


Figura 15. Diagrama de flujo del proceso para cargar ISSN o ISBN.

5.2.5. CREACIÓN DE UNA SOLICITUD DE PUNTUACIÓN DE PRODUCTIVIDAD ACADÉMICA.

En la figura 16 se observa el proceso de solicitud de puntuación de productividad académica, el cual se realiza mediante el diligenciamiento del formulario, brindando la comodidad al docente de desplegar los inputs correspondientes a la modalidad académica seleccionada, con el fin de minimizar errores al momento de la presentación de la solicitud. La validación de los datos ingresados se hace tanto en el front end como en el back end para luego almacenarlos en la base de datos y generar un listado del historial de sus propias solicitudes. Este formulario está basado en el formato MI-INV-FO-05 de la USCO (Anexo 1).

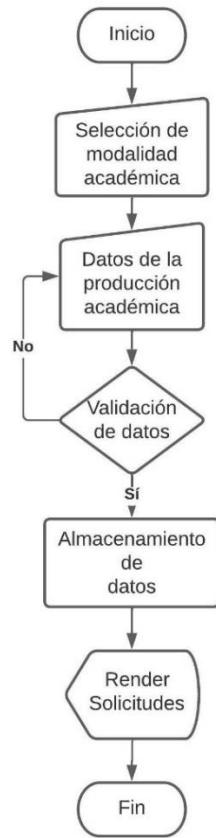


Figura 16. Diagrama de flujo de la creación de una solicitud de puntuación de productividad académica.

5.2.6. GENERACIÓN DE INFORME

En la interfaz del funcionario se encuentra la opción de generar un informe que consta del proceso descrito en la figura 17. Es de forma dinámica ya que tiene tres apartados permitiendo estar o no activos o combinados:

1. Estado y Modalidad: Estos son filtros para la generación de informe que constan del estado en que se encuentran las solicitudes, así como la modalidad académica de estas. Por defecto, todas las casillas de estado se encuentran seleccionadas y el selector de modalidades en la opción “Todas las modalidades académicas”.
2. Año, trimestre y fecha: Esta sección tiene como finalidad dar informes periódicos basándose en estas tres opciones. Si el usuario selecciona un año

y trimestre las fechas de solicitud y publicación quedarán deshabilitadas, si seleccionamos algunas de las fechas las demás casillas de esta sección quedarán deshabilitadas, ya que la búsqueda se acotará en un único rango de fechas.

3. Docente, facultad y programa: Este apartado permite seleccionar un docente en específico, un programa o una facultad de la cual se requiera el informe; A su vez posee un buscador por el número de cédula del docente. Esta búsqueda seleccionará la facultad y el programa del docente, si no se selecciona algunas de estas opciones tomará por defecto todas las posibilidades de este selector. La jerarquía para hacer la búsqueda en esta sección es la siguiente: Docente > Programa > Facultad.

Cabe recalcar que si no se encuentra desplegada alguna de estas tres pestañas o ninguna de ellas, el informe se generará con la mayor cantidad de opciones posibles, ya que si no se encuentran seleccionadas no se aplicará el filtro de esa sección.

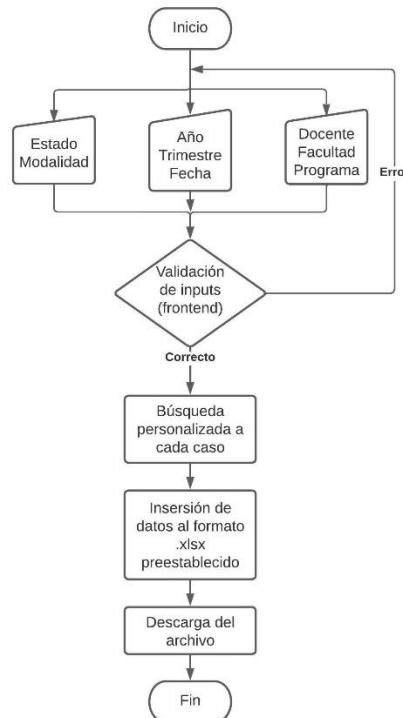


Figura 17. Diagrama de flujo de la generación de informe.

6. DESARROLLO DEL PROYECTO

Para la validación de la plataforma mediante pruebas piloto con usuarios finales, se solicitaron datos a Vicerrectoría de Investigación y Proyección Social.

Dicha solicitud fue negada debido a evidenciado en el anexo 2, por lo tanto, los resultados se realizaron con ejemplos y usuarios ficticios para la simulación de un uso real de la plataforma.

6.1. INTERFAZ DEL ADMINISTRADOR

6.1.1. FORMULARIO PARA AGREGAR USUARIO

En la creación de un usuario (figura18), se encuentran cada uno de los parámetros requeridos para la administración de estos, a través, de toda la plataforma. En el caso de los docentes, se despliegan unas opciones extra necesarias para el informe final.

The screenshot shows a web-based application for managing users. On the left, there's a sidebar with a red header 'Módulo CAP' containing links for 'Inicio', 'Usuarios', 'Números serials', and 'Mi perfil'. Below this is the 'SUCOLOMBIANA' logo. The main content area has a title 'Agregar un nuevo usuario' and a subtitle 'Freddy Leonardo Cuenca Beltrán'. It contains several input fields grouped into sections: 'Información personal' (identification type dropdown, identification number input), 'Datos de localización' (phone input, department dropdown, city dropdown), 'Información de sesión' (teacher dropdown, faculty dropdown, program dropdown), and 'Vinculación' (radio buttons for 'Tiempo completo planta' or 'Medio tiempo planta'). At the bottom right is an 'Agregar' (Add) button. A footer at the bottom right includes the text 'Gupo de Investigación en electrónica, telecomunicaciones e informática - GETI' and '© Universidad Surcolombiana'.

Figura 18. Formulario para agregar un usuario a la plataforma.

6.1.2. LISTADO DE USUARIOS

Este es un listado de todos los usuarios que tiene activos la plataforma en el momento de la realización de las pruebas. Dichos usuarios corresponden a un administrador, un funcionario y cuatro docentes. El administrador podrá ver el perfil, editar o eliminarlo mediante las opciones presentadas en las dos últimas columnas, dicho proceso se observa en la figura 19. A su vez puede realizar

búsquedas mediante la casilla “Buscar”, en donde permite filtrar por cada parámetro que se encuentra en la tabla.

#	Nombres	Primer apellido	Segundo apellido	Identificación	Correo electrónico	Tipo	Perfil	Eliminar
1	Freddy Leonardo	Cuenca	Beltrán	1075265920	leox582@gmail.com	Admin		<button>Eliminar</button>
2	Docente1	Ejemplo	Prueba	112233445	leox582@hotmail.com	Docente		<button>Eliminar</button>
3	Docente 2	Ejemplo 2	Prueba 2	12345678	ejemploDocente2@ejemplo.com	Docente		<button>Eliminar</button>
4	pruebaDocente33	Ejemplo 33	Prueba 33	52146541	pruebaDocente33@protonmail.com	Docente		<button>Eliminar</button>
5	Docente 4	Ejemplo 4	Prueba 4	669988554	ejemploDocente4@ejemplo.com	Docente		<button>Eliminar</button>
6	Funcionario	Ejemplo	Prueba	1232564	ejemploFuncionario1@ejemplo.com	Funcionario		<button>Eliminar</button>

Figura 19. Listado de todos los usuarios.

6.1.3. LISTADO DE LOS ISSN E ISBN

El funcionario tiene la opción de validar mediante la plataforma los seriales de ISSN e ISBN previamente cargados por el administrador.

#	Título	ISSN Impreso	ISSN Electrónico	ISSN L	Instituciones Editoras	Categoría	Vigencia
0	@LIMENTECH	1692-7125	---	---	UNIVERSIDAD DE PAMPLONA	C	2022
1	ACADEMIA & DERECHO	2215-8944	---	---	UNIVERSIDAD LIBRE DE COLOMBIA	B	2022
2	ACADEMIA Y VIRTUALIDAD	---	2011-0731	---	UNIVERSIDAD MILITAR NUEVA GRANADA	C	2022
3	ACADEMIA. REVISTA LATINOAMERICANA DE ADMINISTRACIÓN	1012-8255	---	---	UNIVERSIDAD DE LOS ANDES	C	2022
4	ACTA COLOMBIANA DE PSICOLOGÍA	0123-9155	1909-9711	---	UNIVERSIDAD CATÓLICA DE COLOMBIA	C	2022
5	ACTA MÉDICA COLOMBIANA	0120-2448	2248-6054	---	ASOCIACIÓN COLOMBIANA DE MEDICINA INTERNA - ACMI	C	2022
6	ACTUALIDADES PEDAGÓGICAS	0120-1700	2389-8755	---	UNIVERSIDAD DE LA SALLE	C	2022
7	AD-MINISTER	1692-0279	2256-4322	---	UNIVERSIDAD EAFIT	B	2022
8	ADMINISTRACIÓN & DESARROLLO	0120-3754	2500-5227	0120-3754	ESCUELA SUPERIOR DE ADMINISTRACIÓN PÚBLICA	C	2022
9	AGLALA	---	2215-7360	---	CORPORACION UNIVERSITARIA	C	2022

Figura 20. Listado de seriales ISSN.

Posteriormente, se realiza el cargo de las revistas indexadas homologadas del año 2022 de Colombia tomadas directamente de la página web de Minciencias, el listado se evidencia en la figura 20 indicando las propiedades de cada una, las cuales son:

Nombre de la revista, la institución editora, la categoría siendo A1, A2, B y C, el número de ISSN que puede ser impreso, electrónico o tipo L.

Los seriales ISBN varían para un mismo libro dependiendo la edición emitida, por lo tanto, está basado en el año de publicación. En las pruebas con usuarios finales se cargaron seis seriales (figura 21) usando el formato del anexo 4 con todas sus propiedades ficticias para realizar una comprobación de las solicitudes con modalidad académica de “Libro” o “Capítulo de libro”.

The screenshot shows the 'Módulo CAP' interface with a sidebar on the left containing links: 'Inicio', 'Usuarios', 'Números seriales', and 'Mi perfil'. The main area is titled 'ISBN' and displays a table of six records. The table has columns: '#', 'ISBN', 'Título', 'IMPRESO', 'DIGITAL', 'PROCEDENCIA', 'EDITOR', and 'AÑO DE PUBLICACIÓN'. The data is as follows:

#	ISBN	Título	IMPRESO	DIGITAL	PROCEDENCIA	EDITOR	AÑO DE PUBLICACIÓN
1	1-234-526-8	TITULO DE PRUEBA 1	SI	NO	COLOMBIA	PRUEBA_EDITOR 1	2022
2	3-658-963-7	TITULO DE PRUEBA 2	NO	SI	COLOMBIA	PRUEBA_EDITOR 2	2022
3	8-785-452-4	TITULO DE PRUEBA 3	SI	SI	COLOMBIA	PRUEBA_EDITOR 3	2022
4	6-852-963-8	TITULO DE PRUEBA 4	NO	SI	COLOMBIA	PRUEBA_EDITOR 4	2022
5	5-963-456-7	TITULO DE PRUEBA 5	SI	SI	COLOMBIA	PRUEBA_EDITOR 5	2022
6	4-132-961-6	TITULO DE PRUEBA 6	SI	SI	COLOMBIA	PRUEBA_EDITOR 6	2022

Below the table, there is a footer with the University of Surcolombiana logo and text: 'Ocupo de Investigación en electrónica, telecomunicaciones e informática - OETI' and '© Universidad Surcolombiana'.

Figura 21. Listado de seriales ISBN.

6.2. INTERFAZ DEL DOCENTE

6.2.1. FORMULARIO PARA LA CREACIÓN DE UNA SOLICITUD

El formulario para la solicitud de reconocimiento por productividad académica varía según la modalidad seleccionada. El ejemplo de la figura 22, se observa el formulario más complejo (Artículo de revista), ya que exige una mayor cantidad de datos, como el nombre de la revista, el tipo de artículo, su categoría, serial ISSN, y las fechas tanto de publicación como sometimiento del artículo.

Solicitud de Reconocimiento

Título del Material
Nombre del Artículo

Fecha de Publicación
mm/dd/aaaa

Artículo de Revista

– Tipo de Artículo –

Nombre de la Revista

Categoría de la revista en Mincencias al momento de la publicación

Dirección URL

Seleccionar archivo Ninguno archivo selec.

Tiempo requerido para la publicación del artículo en la revista Cantidad en meses.

Fecha de recepción o sumisión del artículo a la revista mm/dd/aaaa

ISSN

Para la publicación del artículo, se invirtieron recursos económicos de la Universidad Sí NO

Nombre del proyecto de Investigación del cual se genera el material, si aplica

El material presentado fue tenido en cuenta para cambio de categoría en el escalafón docente Sí NO

Número de Autores
1

Enviar

Grupo de investigación en electrónica, telecomunicaciones e informática - GETI
© Universidad Surcolombiana

Figura 22. Formulario para la creación de una solicitud.

6.2.2. LISTADO DE LAS SOLICITUDES DE UN DOCENTE

Todas las solicitudes hechas por el docente se encuentran en la interfaz de este. La figura 23 es un ejemplo de un listado de las solicitudes creadas por un docente con diferentes modalidades académicas, teniendo la posibilidad de ver la fecha de creación de la solicitud, la fecha de publicación de su productividad académica, el título, el URL de la publicación, la opción de descarga del documento adjunto y el estado en el proceso que se encuentra. Con el fin de dar un seguimiento de las solicitudes al docente.

Solicitudes

#	Nombre	Modalidad	URL	Fecha de Solicitud	Fecha de publicación	Estado	Descargar	Eliminar
005	Ejemplo_Estudio posdoctoral	estudios postdoctorales		15/01/2023	09/07/2022	Pendiente por revisión		
004	Ejemplo_Producción software	producción de software		15/01/2023	10/11/2022	Pendiente por revisión		
003	Ejemplo_Revista4	Artículo de Revista		15/01/2023	16/12/2022	Pendiente por revisión		
002	Ejemplo_Revista3	Artículo de Revista		15/01/2023	08/09/2022	Pendiente por revisión		
001	Ejemplo_Libro2	libro		15/01/2023	14/10/2022	Pendiente por revisión		

búsqueda en tabla

Grupo de investigación en electrónica, telecomunicaciones e informática - GETI
© Universidad Surcolombiana

Figura 23. Listado de solicitudes de un docente.

6.3. INTERFAZ DEL FUNCIONARIO

6.3.1. LISTADO DE SOLICITUDES PENDIENTES POR REVISIÓN

El funcionario de la plataforma tiene como responsabilidad, realizar una revisión manual de todos los datos proporcionado por el docente en el momento de la creación de la solicitud, por lo tanto, en su interfaz se encuentra un listado de las solicitudes pendientes por revisión, en donde la última columna le permite dirigirse a la respectiva revisión (figura 24).

#	Docente	Facultad	Programa	Nombre de Publicación	URL	Fecha de Solicitud	Descargar	Revisar
1	pruebaDocente33 Ejemplo 33 Prueba 33	Ciencias Exactas y Naturales	Matemática Aplicada	Ejemplo_Revista1	●	15/01/2023		
2	pruebaDocente33 Ejemplo 33 Prueba 33	Ciencias Exactas y Naturales	Matemática Aplicada	Ejemplo_Revista2	●	15/01/2023		
3	pruebaDocente33 Ejemplo 33 Prueba 33	Ciencias Exactas y Naturales	Matemática Aplicada	Ejemplo_Libro	●	15/01/2023		
4	pruebaDocente33 Ejemplo 33 Prueba 33	Ciencias Exactas y Naturales	Matemática Aplicada	Ejemplo_Patente1	●	15/01/2023		
5	pruebaDocente33 Ejemplo 33 Prueba 33	Ciencias Exactas y Naturales	Matemática Aplicada	Ejemplo_Ponencia1	●	15/01/2023		
6	Docente 2 Ejemplo 2 Prueba 2	Economía y Administración	Contaduría Pública	Ejemplo_Libro2	●	15/01/2023		
7	Docente 2 Ejemplo 2 Prueba 2	Economía y Administración	Contaduría Pública	Ejemplo_Revista3	●	15/01/2023		
8	Docente 2 Ejemplo 2 Prueba 2	Economía y Administración	Contaduría Pública	Ejemplo_Revista4	●	15/01/2023		
9	Docente 2 Ejemplo 2 Prueba 2	Economía y Administración	Contaduría Pública	Ejemplo_Producciónsoftware	●	15/01/2023		
10	Docente 2 Ejemplo 2 Prueba 2	Economía y Administración	Contaduría Pública	Ejemplo_Estudiodpostdoctoral	●	15/01/2023		
11	Docente1 Ejemplo Prueba	Ciencias Exactas y Naturales	Física	Ejemplo_direccióntesis	●	15/01/2023		
12	Docente1 Ejemplo Prueba	Ciencias Exactas y Naturales	Física	Ejemplo_Producióntécnica	●	15/01/2023		

Figura 24. Listado de solicitudes pendientes por revisión.

6.3.2. INTERFAZ DE LA PRIMERA REVISIÓN

Los formularios para la creación de solicitudes por parte de los docentes son adaptados a la modalidad académica seleccionada, de igual manera, en la primera revisión por parte del funcionario las casillas de cada una de las propiedades de la solicitud seleccionada se ajustarán dependiendo de la modalidad seleccionada.

Las opciones “Aceptar”, “Rechazar” o “Editar”, así como el campo para las observaciones si lo requiere, son seleccionadas o diligenciadas por el funcionario, dando paso al siguiente estado de la solicitud. En la figura 25 se observa la interfaz de lo anteriormente dicho.

Figura 25. Interfaz de la primera revisión.

6.3.3. REVISIÓN CAP

El Comité de Asignación de Puntaje somete a evaluación las solicitudes de producción académica que hayan sido aceptadas por el funcionario. Al obtener la respuesta, el funcionario diligencia el formulario “Revisión CAP”, el cual despliega un seleccionador para el tipo de bonificación y una casilla para el puntaje si la opción seleccionada es “Otorgar puntos” (figura 26). Por último, se ingresa el concepto del CAP por el cual se otorgan o no puntos a la solicitud.

Figura 26. Formulario de revisión CAP.

6.4. GENERACIÓN DE INFORME

6.4.1. INTERFAZ DE GENERACIÓN DE INFORME

Esta interfaz está dividida en tres apartados como se explica en el proceso 5.2.6. así como se observa en la figura 27. El funcionario tiene la capacidad de generar un informe con los datos filtrados según se requiera. Cada uno de los apartados posee una validación en el frontend según se seleccionen las opciones deseadas. Por defecto la sección de “Estado y Modalidad” tendrá seleccionadas todas las opciones de estado y todas las modalidades. Para el apartado de “Año, Trimestre y Fecha” la validación está dividida entre Año-Trimestre, Fecha de Solicitud y Fecha de Publicación, habilitando únicamente la opción seleccionada y deshabilitando las otras. Por último, la validación hecha en “Docente, facultad, programa” tiene una jerarquía donde la selección de Docente es el principal filtro, seguido del Programa y por último la Facultad. Cabe aclarar que cada uno de los apartados genera un criterio en el filtro independiente entre sí.

Módulo CAP

Cerrar sesión

Inicio

Solicitudes

Informes

Generar Informe

Perfil

SURCOLOMBIANA

Búsqueda de Solicitudes

Estado y Modalidad

Estado de las Solicitudes

- Pendiente por revisión
- Revisado
- Editar
- Rechazado
- No aprobado por CAP
- Aprobado

Modalidad Académica

Todas las modalidades

Año, Trimestre, Fecha

Año

Trimestre

Fecha de Publicación (Inicio)

dd/mm/aaaa

Fecha de Publicación (Fin)

dd/mm/aaaa

Fecha de Solicitud (Inicio)

dd/mm/aaaa

Fecha de Solicitud (Fin)

dd/mm/aaaa

Docente, Facultad, Programa

Todas las Facultades

Todos los Programas

Ejemplo Docente

Buscar Docente por Cédula

Cédula

Generar

Figura 27. Interfaz de generación de informe.

6.4.2. INFORME

El informe generado por la plataforma (Anexo 5) agrupa todos los datos relevantes con el fin de dar seguimiento a todas las solicitudes presentadas por los docentes de la Universidad Surcolombiana. Dichos datos son ordenados según las experiencias comunicadas por los funcionarios expertos en este proceso.

7. ANÁLISIS DEL DESARROLLO DEL PROYECTO

La plataforma web, ofrece a cada uno de los usuarios (administrador, docente y funcionario) una interfaz amigable, donde el administrador crea, ve y elimina los usuarios de la plataforma, el docente crea, ve y elimina sus solicitudes, y el funcionario da paso a las diferentes revisiones y procesos de las solicitudes que se hayan presentado al comité de asignación de puntaje por parte de los docentes.

Las herramientas implementadas en la plataforma permiten con facilidad almacenar y administrar grandes bancos de datos, así como celeridad y eficacia durante todo el proceso de una solicitud de productividad académica. Toda producción intelectual de la USCO será fácilmente catalogada y ordenada según sean las necesidades de los entes auditores de estas.

Con el fin de comprobar los ISSN e ISBN diligenciados por los docentes al momento de la creación de la solicitud de productividad académica es necesario realizar un cargue masivo de los seriales actualizados a nivel nacional e internacional tanto de revistas como de libros publicados, con el objetivo de obtener agilidad en el proceso y eficiencia de la finalidad de la plataforma web, ya que anteriormente este proceso se realizaba de forma manual, consultando cada uno de los números en las plataformas dedicadas a la recolección de dicha información, haciendo el proceso mas largo que el propuesto en este proyecto.

Finalmente mediante las opciones ofrecidas por los frameworks y módulos de NodeJS, se obtuvieron soluciones que se adaptan claramente a las necesidades propuestas por la comunidad académica, siendo esta una plataforma totalmente escalable para futuros desarrollos o adaptaciones a herramientas existentes en la Universidad Surcolombiana.

8. CONCLUSIONES

Los lenguajes HTML, CSS y principalmente Javascript hacen de esta plataforma totalmente escalable y adaptable a cualquier navegador web, para ser amigable, clara y ordenada en su uso.

Las tecnologías NodeJS, ExpressJS, JQuery y MongoDB son la base para el desarrollo, diseño e implementación de esta plataforma, que logra dar administración, seguimiento e informes de la producción intelectual de los docentes de la Universidad Surcolombiana.

Debido a las restricciones ante la ley 1581 del 2012 las pruebas finales de la plataforma se realizaron con datos ficticios. Sin embargo, la plataforma fue desarrollada bajo el seguimiento de usuarios finales de esta, dando como aprobado y satisfactorio los resultados de la misma; obteniendo así una plataforma web capaz de administrar la producción intelectual de los docentes de la Universidad Surcolombiana.

9. RECOMENDACIONES

Para futuras líneas de trabajo realizadas sobre este software, es necesario el desarrollo de la seguridad informática, ya que la información administrada por la plataforma es confidencial. En adición la finalidad de este proyecto es ser adaptado a todas las diferentes dependencias de la universidad, así como a la propia plataforma web de la institución.

10. BIBLIOGRAFÍA

BIBEAULT, Bear; DE ROSA, Aurelio y KATZ, Yehuda. *jQuery in Action*. Third edition. Manning. 2015. ISBN: 9781617292071.

BODIN, Martin *et al*. A trusted mechanised JavaScript specification. ACM SIGPLAN Notices, 49(1), 87-100. 2014.

CANTELON, Mike *et al*. *Node. js in Action*. Greenwich: Manning. p. 17-20. 2014. ISBN 9781617290572.

CEDERHOLM, Dan y ZELDMANN, Jeffrey. *CSS3 for web designers*. Second edition. New York, NY: A Book Apart. 2010. p. 56. ISBN: 9780984442522.

CORRADI, Valentina y SWANSON, R Norman. Bootstrap specification tests for diffusion. 2005. ISBN: 0304-4076.

EYADA, Mahmoud M *et al*. Performance evaluation of iot data management using mongodb versus mysql databases in different cloud environments. Vol 8. IEEE Access. 2020. DOI: 10.1109/ACCESO.2020.3002164.

GASSTON, Peter. *The book of CSS3: A Developer's Guide to the Future of Web Design*. no starch press. 2011.

HAHN, Evan M. *Express in Action: Writing, building, and testing Node. js applications*. Simon and Schuster. 2016. ISBN 9781617292422.

SEONGHOON, Kang y SUKYOUNG, Ryu. Formal specification of a JavaScript module system. In Proceedings of the ACM international conference on Object oriented programming systems languages and applications. 2012. p. 621-638. ISBN: 0362-1340.

KRÓL, Karol y SZOMOROVA, Lenka. The possibilities of using chosen jQuery JavaScript components in creating interactive maps. Geomatics, Landmanagement and Landscape, (2), 45-54. 2015. <http://dx.doi.org/10.15576/gll/2015.2.45>.

RAGGETT, David; LE HORS, Arnaud y JACOBS, Ian. HTML 4.01 Specification. W3C, 1999. recommendation, 24.

TEIXEIRA, Pedro. Professional Node.js: Building Javascript based scalable software. John Wiley & Sons. 2012. ISBN: 9781118185469.

ANEXOS

Anexo 1. Formato MI-INV-FO-05.

	UNIVERSIDAD SURCOLOMBIANA INVESTIGACIÓN				
SOLICITUD PUNTUACIÓN DE PRODUCTIVIDAD ACADÉMICA					
CÓDIGO	MI-INV-FO-05	VERSIÓN	8	VIGENCIA	2022
			PÁGINA	1 de 2	

I. Información del Solicitante

No. Identificación			
Nombres y Apellidos			
Facultad			
Programa			
Vinculación docente	Tiempo completo planta	Medio tiempo planta	

Pertenece a un grupo de investigación? Sí No

Si su respuesta es afirmativa, favor complemente las siguientes preguntas:

Nombre del grupo			
Línea de investigación			

II. Tipo de material presentado para puntuación: Marque con una (X)

01 Artículo de Revista	02 Video de Carácter Internacional	03 Video de Carácter Nacional	04 Libro derivado de Investigación	05 Libro de Texto	06 Libro de Ensayo	07 Premios Nacionales	08 Premios Internacionales	09 Patentes	10 Traducciones	11 Obras Artísticas	12 Producción Técnica	13 Producción de Software	14 Ponencia Nacional	15 Ponencia Internacional	16 Publicaciones Impresas Universitarias	17 Estudios Postdoctorales	18 Reseñas Críticas	19 Direcciones de Tesis	20 Capítulo de libro
------------------------	------------------------------------	-------------------------------	------------------------------------	-------------------	--------------------	-----------------------	----------------------------	-------------	-----------------	---------------------	-----------------------	---------------------------	----------------------	---------------------------	--	----------------------------	---------------------	-------------------------	----------------------

Título del Material:

III. La siguiente información, solo deberá ser diligenciada si el material presentado es artículo de revista:

Nombre de la revista: _____

Tipo de Artículo:

Investigación	Comunicación corta	Revisión	Reflexión	Reporte de caso
---------------	--------------------	----------	-----------	-----------------

ISSN: _____ Categoría de la revista en Minciencias al momento de la publicación: _____

Dirección URL: _____

Tiempo requerido para la publicación del artículo en la revista: _____ Meses

Fecha de recepción o sumisión del artículo a la revista: _____

Fecha de aprobación y/o publicación del artículo: _____

Para la publicación del artículo, se invirtieron recursos económicos de la Universidad: Sí No _____

Vigilada Mineducación
La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebidamente no es de responsabilidad de la Universidad Surcolombiana

	UNIVERSIDAD SURCOLOMBIANA INVESTIGACIÓN				
SOLICITUD PUNTUACIÓN DE PRODUCTIVIDAD ACADÉMICA					
CÓDIGO	MI-INV-FO-05	VERSIÓN	8	VIGENCIA	2022
			PÁGINA	2 de 2	

IV. La siguiente información, solo deberá ser diligenciada si el material presentado es libro o capítulo de libro:

Tipo de libro o capítulo de libro: Investigación: Texto: Ensayo: _____

ISBN: _____ Editorial: _____

Dirección URL Editorial: _____

V. Información producción académica (La siguiente información solo deberá ser diligenciada, si el tipo de material seleccionado por usted es: 01, 04, 05, 06, 09, 14, 15, 20)

Nombre del proyecto de Investigación del cual se genera el material, si aplica: _____

VI. El material presentado fue tenido en cuenta para cambio de categoría en el escalafón docente:

SI NO

VII. Número de ejemplares presentados¹

VIII. Declaración de responsabilidad de la información:

En mi calidad de docente de la Universidad Surcolombiana de Neiva Huila, me obligo a hacer envío del presente formato debidamente diligenciado, con los soportes correspondientes. Me abstengo de enviar información que no sea real, y/o que no esté debidamente soportada por los documentos idóneos para este efecto. En caso de no tener certeza acerca de la veracidad de la información enviada al Comité Asignación de Puntaje - CAP, o de no tener seguridad en cuanto a los soportes idóneos para sustentar estos datos, me abstendré de realizar el registro.
En estos términos declaro mi responsabilidad sobre el manejo que daré a la información personal y académica a enviar al CAP.

IX. Información de recepción (Este espacio solo debe ser diligenciado por la Vicerrectoría de Investigación y Proyección Social).

Fecha de recibido en la Vicerrectoría de Investigación y Proyección Social:	Firma del Profesor(a) Solicitante
Fecha de recibido en el CAP:	

¹ Los tipos de material que requieren presentar dos (2) ejemplares, son los identificados con los numerales: 04, 05 y 06.

Vigilada Mineducación
La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebidamente no es de responsabilidad de la Universidad Surcolombiana

Anexo 2. Carta respuesta de Vicerrectoría de Investigación y Proyección Social para solicitud de información para pruebas finales.



MEMORANDO No. 605

Neiva, 2 de noviembre de 2022

PARA: **ALBEIRO CORTÉS CABEZAS**
Docente Planta
Programa de Ingeniería Electrónica

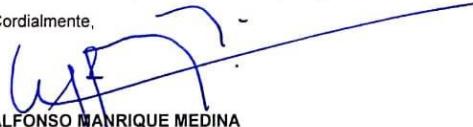
DE: **ALFONSO MANRIQUE MEDINA**
Vicerrector de Investigación y Proyección Social
Presidente Comité de Asignación de Puntaje – CAP

Ref. Respuesta oficio de 25 de octubre de 2022.

Respetado Doctor.

En referencia al oficio del 25 de octubre de 2022, en donde solicita el archivo en Excel de productividad académica de los últimos 5 años para el desarrollo del proyecto de grado titulado "Diseño e Implementación de una Plataforma para la Administración de la Producción Intelectual de los Docentes de la Universidad Surcolombiana", de manera respetuosa se informa que, teniendo en cuenta las disposiciones de la Ley 1581 de 2012, sobre la protección de datos personales, el Comité de Asignación de Puntaje recomendó que, a partir del año 2018, no se deben volver a dar información de la producción académica ni publicar en la página web de la Universidad las actas de formal reconocimiento de puntos, en aras de proteger la intimidad de cada docente.

Cordialmente,


ALFONSO MANRIQUE MEDINA
Vicerrector de Investigación y Proyección Social
Presidente Comité de Asignación de Puntaje – CAP

Proyecto: *Luis Lizardo Santafé Tór*
Secretaria CAP


2-11-22

VICERRECTORÍA DE INVESTIGACIÓN Y PROYECCIÓN SOCIAL

• Sede Central / Av. Pastrana Borrero, Cra. 1 • PBX. 875 4753
• Sede Administrativa / Cra. 5 No. 23 - 40 • PBX. 875 3686
• www.usco.edu.co / Neiva - Huila • Línea Gratuita Nacional: 018000 988722



 Escaneado con CamScanner

Anexo 3. Formato .xlsx para cargar ISSN.

A No.	B TÍTULO	C ISSN IMPRESO	D ISSN ELECTRÓNICO	E ISSN L	F INSTITUCIONES EDITORAS	G CATEGORÍA	H VIGENCIA
1							
2	1						
3	2						
4	3						
5	4						
6	5						
7	6						
8	7						
9	8						
10	9						
11	10						
12	11						
13	12						
14	13						
15	14						
16	15						
17	16						
18	17						
19	18						
20	19						
21	20						

Anexo 4. Formato .xlsx para cargar ISBN.

A No.	B TÍTULO	C IMPRESO	D ELECTRÓNICO	E EDITORIAL	F AÑO DE PUBLICACIÓN
1					
2					
3					
4					
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					

Anexo 5. Informe general.

A	B	C	D	E	F	G	H	I
No.	FECHA DE SOLICITUD	VIGENCIA	FACULTAD	PROGRAMA	IDENTIFICACIÓN	NOMBRES COMPLETOS	TIPO DE MATERIAL	TÍTULO DEL MATERIAL
1								
2	1 15/01/2023	2023	Ciencias Exactas y Naturales	Matemática Aplicada	52146541	pruebaDocente33 Ejemplo 33	Artículo de Revista	Ejemplo_Revista1
3	2 15/01/2023	2023	Ciencias Exactas y Naturales	Matemática Aplicada	52146541	pruebaDocente33 Ejemplo 33	Artículo de Revista	Ejemplo_Revista2
4	3 15/01/2023	2023	Ciencias Exactas y Naturales	Matemática Aplicada	52146541	pruebaDocente33 Ejemplo 33	Libro	Ejemplo_Libro
5	4 15/01/2023	2023	Ciencias Exactas y Naturales	Matemática Aplicada	52146541	pruebaDocente33 Ejemplo 33	Patentes	Ejemplo_Patente1
6	5 15/01/2023	2023	Ciencias Exactas y Naturales	Matemática Aplicada	52146541	pruebaDocente33 Ejemplo 33	Ponencia	Ejemplo_Ponencia1
7	6 15/01/2023	2023	Economía y Administración	Contaduría Pública	12345678	Docente 2 Ejemplo 2	Libro	Ejemplo_Libro2
8	7 15/01/2023	2023	Economía y Administración	Contaduría Pública	12345678	Docente 2 Ejemplo 2	Artículo de Revista	Ejemplo_Revista3
9	8 15/01/2023	2023	Economía y Administración	Contaduría Pública	12345678	Docente 2 Ejemplo 2	Artículo de Revista	Ejemplo_Revista4
10	9 15/01/2023	2023	Economía y Administración	Contaduría Pública	12345678	Docente 2 Ejemplo 2	Producción de software	Ejemplo_Produccionsoftware
11	10 15/01/2023	2023	Economía y Administración	Contaduría Pública	12345678	Docente 2 Ejemplo 2	Estudios postdoctorales	Ejemplo_Estudiodoctoral
12	11 15/01/2023	2023	Ciencias Exactas y Naturales	Física	112233445	Docente1 Ejemplo	Direcciones de tesis	Ejemplo_direccióntesis
13	12 15/01/2023	2023	Ciencias Exactas y Naturales	Física	112233445	Docente1 Ejemplo	Producción técnica	Ejemplo_Produccióntécnica
14	13 15/01/2023	2023	Ciencias Exactas y Naturales	Física	112233445	Docente1 Ejemplo	Libro	Ejemplo_Libro1
15	14 15/01/2023	2023	Ciencias Exactas y Naturales	Física	112233445	Docente1 Ejemplo	Artículo de Revista	Ejemplo_Revista6
16	15 15/01/2023	2023	Ciencias Exactas y Naturales	Física	112233445	Docente1 Ejemplo	Artículo de Revista	Ejemplo_Revista7
17	16 15/01/2023	2023	Ingeniería	Ingeniería de Petróleos	669988554	Docente 4 Ejemplo 4	Libro	Ejemplo_Libro
18	17 15/01/2023	2023	Ingeniería	Ingeniería de Petróleos	669988554	Docente 4 Ejemplo 4	Libro	Ejemplo_Libro2
19	18 15/01/2023	2023	Ingeniería	Ingeniería de Petróleos	669988554	Docente 4 Ejemplo 4	Artículo de Revista	Ejemplo_Revista8
20	19 15/01/2023	2023	Ingeniería	Ingeniería de Petróleos	669988554	Docente 4 Ejemplo 4	Artículo de Revista	Ejemplo_Revista9
21	20 15/01/2023	2023	Ingeniería	Ingeniería de Petróleos	669988554	Docente 4 Ejemplo 4	Libro	Ejemplo_Libro4

Hoja1

J	K	L	M	N	O	P	Q	R	S	T	U
TRADUCCIÓN TÍTULO DE REVISTA	PUBLICADO Y/O PARTICIPACIÓN EN	EDITORIAL	FECHA DE PUBLICACIÓN O REALIZACIÓN	ISBN & ISSN	No AUTORES	AUTORES	CATEGORÍA REVISTA	CONCEPTO DEL CAP	PUNTOS	TIPO DE PUNTO	Referenciado en Scopus SI/NO
AGRONOMÍA COLOMBIANA			30/09/2022	0120-9965	1	Ejemplo_Autor2;Ejemplo_Autor3	B	Ejemplo_Otorgar Puntos Ejemplo_Revista1	3	Reconocimiento de puntos salariales Bonificación por productividad	
AVANCES EN ENFERMERÍA	Ejemplo_editorial		09/11/2022	2346-0261	3	Ejemplo_Autor2	B	Ejemplo_Otorgar Puntos Ejemplo_Revista2	12	académica	
			15/07/2022	1-234-526-8	2	Ejemplo_Autor2	N/A				
			16/11/2022		2	Ejemplo_Autor2	N/A				
			10/11/2022		1	N/A					
		Ejemplo_Editorial2	14/10/2022	3-658-963-7	2	Ejemplo_Autor2	N/A				
ANTÍPODA BIOMÉDICA			08/09/2022	1900-5407	1	N/A					
			16/12/2022	2590-7379	2	Ejemplo_Autor2	B				
			10/11/2022		2	Ejemplo_Autor2	N/A				
			07/09/2022		1	N/A					
			08/09/2022		1	N/A					
		Ejemplo_Editorial	10/11/2022	8-785-452-4	2	Ejemplo_Autor2	N/A	Ejemplo_Otorgar Puntos Ejemplo_Produccióntécnica	6	Reconocimiento de puntos salariales	
CIENCIA Y TECNOLOGÍA AGROPECUARIA	Ejemplo_Editorial		03/11/2022		2	Ejemplo_Autor2	N/A				
			13/10/2022	2500-5907	1	N/A					
CALDASIA	Ejemplo_Editorial		02/11/2022	0366-5232	1	Ejemplo_Autor2;Ejemplo_Autor3	C	Ejemplo_Puntos Denegados			
			01/11/2022	5-963-456-7	2	Ejemplo_Autor2	B				
		Ejemplo_Editorial7	28/09/2022	4-132-962-3	2	Ejemplo_Autor2	N/A	Ejemplo_Puntos Denegados			
BOLETÍN DE INVESTIGACIONES MARINAS Y COSTERAS			29/09/2022	2590-4682	1	Ejemplo_Autor2;Ejemplo_Autor3	C				
CIENCIA EN DESARROLLO	Ejemplo_Editorial5		18/11/2022	0121-7488	3	Ejemplo_Autor2;Ejemplo_Autor3	B				
			14/07/2022	5-963-456-7	2	Ejemplo_Autor2	N/A	Ejemplo_Puntos Denegados Ejemplo_Libro4			

MÓDULO CAP



**MANUAL DEL
PROGRAMADOR**

ARQUITECTURA

Freddy Leonardo Cuenca Beltrán

Plataforma web para la administración de la producción intelectual de los docentes de la Universidad Surcolombiana.

TABLA DE CONTENIDO

DESCRIPCIÓN.....	3
1. Login	4
ADMINISTRADOR.....	3
1. Primer usuario administrador de la plataforma.....	5
2. Agregar usuario.....	5
3. Seriales ISSN e ISBN.....	6
DOCENTE	8
1. Creación de una solicitud.....	8
2. Administración de las solicitudes	9
FUNCIONARIO.....	10
1. Listado de solicitudes pendientes por revisión.....	10
2. Interfaz para la primera revisión.....	10
3. Revisión CAP.....	10
4. Interfaz de generación de informe.....	10

DESCRIPCIÓN

Este manual describe a detalle la arquitectura, la base de datos y las parejas de rutas/controladores la plataforma. Especificando concretamente las herramientas utilizadas, módulos, procesos, modelos y apartados. Los lenguajes utilizados son HTML y Javascript, este último con la peculiaridad de su uso en el frontend y en el backend de este aplicativo web, facilitando su entendimiento y escalabilidad. El enfoque utilizado en este manual es la explicación de los procesos de las funciones y secuencias que dan paso a todas las acciones ejecutadas por los usuarios de la plataforma.

ARQUITECTURA.

1. MODELO VISTA CONTROLADOR

La arquitectura de la plataforma está basada en el estilo Modelo Vista Controlador (MVC)(Figura1) con enfoque en un CRUD de los datos del aplicativo.

La gestión de datos CRUD(Create, Read, Update, Delete) está dada en toda la plataforma. Las peticiones HTTP get en los enrutadores se usan para renderizar vistas, post para crear datos mediante formularios, put para editar datos y delete para eliminarlos. Estas acciones son ejecutadas por los controladores de las rutas.

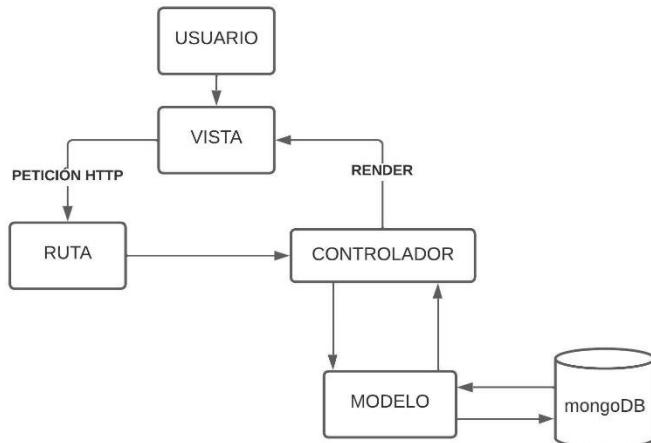


Figura 1. Modelo vista controlador.

2. FRAMEWORKS Y MÓDULOS

El framework usado para esta arquitectura es ExpressJS (Figura 2), el más popular y utilizado de NodeJS para la creación de aplicativos webs.

```
1 const express = require('express')
2 const app = express()
3 const port = 3000
4
5 app.get('/', (req, res) => {
6   res.send('Hello World!')
7 })
8
9 app.listen(port, () => {
10   console.log(`Example app listening on port ${port}`)
11 })
```

Figura 2. Ejemplo estandarizado de “Hola mundo” de Express.js mediante una

petición HTTP GET.

El motor de vistas (view engine) es Express Handlebars (Figura 3) está enfocado en ser el motor de vistas de ExpressJS.

```
const app = express();
const hbs = create({ /* config */ });

app.engine('handlebars', hbs.engine);
app.set('view engine', 'handlebars');
app.set('views', './views');
```

Figura 3. Modelo estandarizado de la configuración del motor de vistas.

Los módulos de autenticación y sesión son passport-local y express-session. (Figura 4).

```
const session = require('express-session');
const passport = require('passport');

app.use(session({
  secret: 'secret',
  resave: true,
  saveUninitialized: true
}));
app.use(passport.initialize());
app.use(passport.session());
```

Figura 4. Ejemplo estandarizado para la configuración de passport y express-session.

Los mensajes guardados en la sesión del usuario y usados para dar alertas desde el servidor junto a una redirección son ejecutados con el módulo "connect-flash" basado en "ExpressJS" (Figura 5).

```

var flash = require('connect-flash');
var app = express();

app.configure(function() {
  app.use(express.cookieParser('keyboard cat'));
  app.use(express.session({ cookie: { maxAge: 60000 } }));
  app.use(flash());
});

```

Figura 5. Ejemplo estandarizado de connect-flash de ExpressJS.

3. PUESTA EN MARCHA.

Ya configurado el servidor con sus módulos, arquitectura y conexión con la base de datos, se pone en marcha la plataforma ejecutando el archivo index.js (Figura 6). Este archivo es el ejecutable para futuros despliegues de la plataforma usando algún gestor de procesos de producción de NodeJS como pm2. En desarrollo se recomienda usar el módulo nodemon.

Los envíos de correos se hacen mediante el módulo NodeMailer de NodeJS. Es necesario para esto configurar un correo para el uso de aplicaciones externas e ingresar estas credenciales en cada uno de los usos de este módulo en la plataforma.

```

require('dotenv').config();
const app = require('./server');
require('./database');

app.listen(app.get('port'), () => {
  console.log('Server on port:', app.get('port'))
})

```

Figura 6. Inicialización de la plataforma.

4. CDN

CDN (Content Delivery Network o Red de Distribución de Contenido) Varias herramientas que usamos en la plataforma están implementados por CDN para el uso en el Frontend.

Se hace referencia a los CDN ya que tienen gran importancia en el funcionamiento

de la plataforma, especialmente en el archivo functions.js.

El archivo functions.js hace un amplio uso de la biblioteca jQuery, con el fin de facilitar todo el manejo del DOM en la plataforma, así como para realizar peticiones en el frontend mediante AJAX.

Bootstrap es un framework front-end utilizado para dar estilos y etiquetas que ayudan a hacer amigable y responsive la interfaz del usuario.

Los íconos usados en la plataforma son proporcionados por Fontawesome que es un framework de íconos vectoriales.

Las alertas emitidas por el frontend son emitidas por SweetAlert2 que es un plugin de JQuery.

Finalmente, Datatables permite realizar búsquedas en las tablas generadas en toda la plataforma.

BASE DE DATOS

1. CONEXIÓN

La conexión y manejo de la base de datos (MongoDB) se hace mediante el módulo mongoose (Figura 7).

```
const mongoose = require('mongoose')

let PID_APP_HOST='127.0.0.1'
let PID_APP_DATABASE='PID_DB'

const MONGODB_URI = `mongodb://${PID_APP_HOST}/${PID_APP_DATABASE}`;

mongoose.connect(MONGODB_URI, {
  useNewUrlParser: true,
  useUnifiedTopology: true
})
.then(db => console.log(`Database is connected`))
.catch(err => console.log(err));
```

Figura 7. Conexión a la base de datos (database.js).

Esta configuración está basada en la documentación del módulo. Como principales características de este código está la dirección del host, clave para una configuración en la nube, el nombre de la base de datos, necesaria para usar el Shell de Mongo y la configuración de la función “connect”.

2. MODELOS

A lo largo del aplicativo web es de tener en cuenta la estructura de cada uno de los modelos usados. La carpeta Models, contiene estos archivos.

2.1. USERS

para los tres roles de usuario: Admin, Funcionario y Docente. Los tres comparten el mismo Schema diferenciado por el valor en la propiedad role. (Figura 8).

```
const UserSchema = Schema({
  identification_type: {type: String},
  identification: { type: String, required: true, unique: true},
  name: {type: String, required: true},
  lastname: {type: String, required: true},
  sec_lastname: {type: String},
  role: {type: String, required: true, enum: ["Admin", "Funcionario", "Docente"]},
  email: {type: String, required: true, unique: true},
  password: {type: String, required: true},
  facultad: {type: String},
  programa: {type: String},
  vinculacion: {type: Boolean},
  investigacion: {type: Boolean},
  nombreGrupo: {type: String},
  lineaInvestigacion: {type: String},
  last_login_date: {type: Date, default: Date.now()}
}, {timestamps: true});
```

Figura 8. Modelo Schema de usuario (Users.js)

Las propiedades principales de este modelo son las que se encuentran requeridas (required: true), siendo el número de identificación, el nombre completo, role, usuario y contraseña, estos dos últimos son los datos validados en el momento del inicio de la sesión mediante el módulo Passport.

Las variables no requeridas facultad, programa, vinculacion, investigacion, nombreGrupo, lineaInvestigacion serán *null* para el usuario Admin, así como para el Funcionario.

Los docentes tienen la propiedad facultad y programa para futuros filtros. Las propiedades de vinculacion, investigacion, nombreGrupo, lineaInvestigacion, son exclusivos del usuario Docente.

2.2. SOLICITUDES

Esta colección guarda las solicitudes creadas por los docentes (Figura 9) y el estado en el que se encuentra la solicitud que consta de seis estados:

1. “Pendiente por revisión”: En este estado la solicitud esta recién creada por el docente y puesta en lista de pendiente por revisión del funcionario.

2. “Revisado”: El funcionario ha aceptado la solicitud para presentarla ante el CAP.
3. “Editar”: El funcionario ha decidido devolver la solicitud con la opción de modificar la solicitud junto a una observación especificando las correcciones a efectuar.
4. “Rechazado”: Cuando una solicitud se encuentra con datos erróneos sin opción de corregir, el funcionario podrá rechazar totalmente esta solicitud y el docente deberá crear una nueva.
5. “No aprobado por CAP”: Una solicitud validada por el funcionario se someterá ante el CAP. Si este comité decide no otorgar puntos por productividad académica la solicitud se encontrará en estado “No aprobado por CAP”, junto al concepto emitido por él.
6. “Aprobado”: En caso de que el CAP otorgue puntos a la productividad académica, el estado de esta solicitud será “Aprobado”, podrá dar valor a la propiedad “puntaje”, qué tipo de puntaje (true = Bonificación por productividad académica o false = Reconocimiento de puntos salariales) y el concepto por el cual se aprobó.

```
const PublicationSchema = Schema({
  id_Docente: Schema.Types.ObjectId,
  name: {type: String, required: true},
  modalidad: {type: String, required: true},
  categoria: {type: String},
  tipo: {type: String},
  nombre_revista:{type: String},
  tiempo_revista:{type: Number},
  fecha_recepcion_revista:{type: Date},
  fecha_publicacion: {type: Date},
  ISBN:{type:String}, //ISSN ó ISBN
  recursos_U:{type:Boolean},
  nombre_proyecto_investigacion:{type: String},
  editorial:{type:String},
  URL: {type: String},
  cambio_categoria: {type: Boolean},
  //Datos del Archivo
  filename: {type: String},
  path: {type: String},
  originalname: {type: String},
  mimetype: {type: String},
  size: {type: Number},
  //Revisiones
  estado: {type: String,enum: ['Pendiente por revisión','Revisado', 'Editar',
  'Rechazado','No aprobado por CAP','Aprobado'],
  default: 'Pendiente por revisión'},
  observacion:{type: String},
  conceptoCAP:{type: String},
  tipo_puntaje:{type: Boolean},
  puntaje: {type: Number}
},{timestamps: true});
```

Figura 9. Modelo Schema de una solicitud.

La propiedad “ISBN” almacena el valor del ISSN o ISBN para las diferentes modalidades que lo requieran, siendo null para las que no dependan de una revista indexada o libro.

Las propiedades “modalidad”, “categoría” y “tipo”, varían según la clasificación que

tenga la modalidad, por ejemplo, la modalidad “Artículo de Revista” posee tipo de investigación, comunicación corta, revisión, reflexión o reporte de caso, y la revista en las categorías de A1, A2, B y C; en cambio una “Producción de videos, cinematográficas o fonográficas” tiene impacto internacional o nacional siendo este dato almacenado en tipo y la finalidad de la producción didáctico o documental almacenado como categoría. Quienes no posean tipo o categoría, su valor será null. Las propiedades que corresponden a los datos del archivo (filename, path, originalname, mimetype, size) tienen como finalidad el almacenamiento de los archivos cargados por los docentes en la solicitud.

2.3. MODELO DE AUTOR (autor.js)

Una productividad académica puede tener múltiples autores, por lo tanto, es una buena solución poseer una colección que brinde la opción de ser dinámica esta variable en una solicitud. La figura 10 muestra que las propiedades del modelo son “id_publication” que será la propiedad que almacena el indicador único de la solicitud con la cual está relacionada y el nombre del autor.

```
const Autor = Schema({
  id_publication: Schema.Types.ObjectId,
  nombre: {
    type: String,
    required: true
  },
  {
    timestamps: true
  });

```

Figura 10. Modelo Schema de un autor de una solicitud.

2.4. MODELO DE INTERNATIONAL STANDARD SERIAL NUMBER – NÚMERO INTERNACIONAL NORMALIZADO DE PUBLICACIONES SERIADAS (ISSN) (ISSN.js)

La plataforma permite cargar masivamente los datos de las revistas indexadas mediante un archivo de formato .xlsx con las características requeridas para dar una verificación de los números proporcionados por los docentes en las solicitudes que contengan estos seriales.

Actualmente un ISSN puede ser impreso, electrónico o tipo L. Si se desea cargar un serial de años anteriores que no posean estas descripciones, se almacenará este número en la propiedad “issn_impreso” por defecto. (Figura 11).

```

const ISSN = Schema({
    titulo: {type: String, required: true},
    issn_impreso:{type: String},
    issn_electronico:{type: String},
    issn_l:{type: String},
    institucion_editora:{type: String},
    categoria:{type:String, required: true},
    vigencia:{type: Number, required: true}
},{
    timestamps: true
});

```

Figura 11. Modelo Schema de ISSN.

2.5. MODELO DE INTERNATIONAL STANDARD BOOK NUMBER - CÓDIGO INTERNACIONAL NORMALIZADO PARA LIBROS (ISBN) (ISBN.js)

Las modalidades de productividad académica “Libro” y “Capítulo del libro”, tienen como propiedad el número serial correspondiente al libro de la productividad.

Las propiedades que definen este serial son título, si es un obra impresa o digital, lugar de procedencia y el numero serial del editor visualizados en la figura 12. Esta colección permite almacenar masivamente los ISBN para poder validar en la interfaz del funcionario cada serial.

```

const ISBN = Schema({
    ISBN:{type: String, required: true},
    titulo:{type: String, required: true},
    impreso:{type: Boolean},
    digital:{type: Boolean},
    procedencia:{type: String},
    editor:{type:String},
    yyyy_publicacion:{type:Number}
},{
    timestamps: true
});

```

Figura 12. Modelo Schema de ISBN.

RUTAS Y CONTROLADORES

Toda ruta está estrechamente relacionada con un controlador. Cada archivo de ruta tiene un archivo controlador, con el fin de separar cada funcionalidad de la plataforma. Todas las rutas se encuentran en la carpeta routes y los controladores en la carpeta controllers. La excepción a esto es el archivo passport.js (figura 13) en la carpeta config, debido a la configuración dada por la documentación de este módulo con la estrategia passport-local. Cabe resaltar que el módulo de passport puede tener otras estrategias como passport-facebook, passport-google-auth, passport-twitter, entre otras.

```

const passport = require('passport');
const LocalStrategy = require('passport-local').Strategy;

const User = require('../models/User');

passport.use(new LocalStrategy({ ...
}));

passport.serializeUser((user, done) => { ...
});

passport.deserializeUser((id, done) => { ...
});

```

Figura 13. Configuración de la estrategia passport-local.

La última excepción a la dualidad ruta controlador es el archivo auth.js (figura 14) en la carpeta helpers teniendo como uso la validación de la autenticación en la plataforma y comprobación de role en cualquier petición HTTP ejecutándose entre la secuencia ruta y el controlador.

```

const helpers = {};

helpers.isAuthenticated = (req, res, next) => { ... };

helpers.isFuncionario = (req, res, next) => { ... };

helpers.isAdmin = (req, res, next) => { ... };

helpers.isDocente = (req, res, next) => { ... };

module.exports = helpers;

```

Figura 14. Controladores de funciones de validación de usuario.

Las validaciones de usuario poseen la entrada next para una vez hecha la comprobación continúe con la ejecución de la siguiente función, por lo general un controlador.

Los controladores isFuncionario, isAdmin e isDocente, son las comprobaciones de cada uno de los roles de la plataforma. isAuthenticated comprueba la sesión activa del usuario.

1. INDEX

La pareja index.routes.js (figura 15) e index.controller.js (figura 16) tiene como uso el renderizado de home.hbs para cada uno de los usuarios. Aunque se use un único controlador para esto, las condicionales permitidas en los archivos .hbs por el motor de vistas express-handlebar permite dar diferenciación a las vistas de los usuarios mediante código HTML. A esta funcionalidad del motor de vistas se da uso a lo largo de la plataforma.

```
const {Router} = require('express')
const router = Router();

const {renderIndex} = require('../controllers/index.controllers')

router.get('/', renderIndex);

module.exports = router;
```

Figura 14. Enrutador index.routes.js.

```
const indexCtrl = {};

indexCtrl.renderIndex = (req, res) => { ... };

module.exports = indexCtrl;
```

Figura 16. Controlador index.controller.js.

2. USERS

Todos los procesos referentes a los usuarios se encuentran en las rutas users.routes.js (figura17) y users.controllers.js (figura 18).

```

const {Router} = require('express')
const router = Router();
const {isAuthenticated, isAdmin} = require('../helpers/auth');
const ...  

    myProfile} = require('../controllers/users.controllers')  

//Ingreso a la plataforma  

router.get('/users/signInForm', signInForm);
router.post('/users/signIn', signIn);
router.get('/users/signInError', signInError);
router.get('/users/updateLogin', isAuthenticated, updateLogin);
router.get('/users/home', isAuthenticated, home);
router.get('/users/logout', logout);
//Acciones del admin  

router.get('/users/createUserForm', isAuthenticated, isAdmin, createUserForm);
router.post('/users/createUser', isAuthenticated, isAdmin, createUser);
router.get('/users/createAdminForm', createAdminForm);
router.post('/users/createAdmin', createAdmin);
router.get('/users/seeAllUsersAdmin', isAuthenticated, isAdmin, seeAllUsersAdmin);
router.get('/users/seeUserAdminForm/:id', isAuthenticated, isAdmin, seeUserAdminForm);
router.get('/users/editUserFormAdmin/:id', isAuthenticated, isAdmin, editUserFormAdmin);
router.put('/users/editUserAdmin/:id', isAuthenticated, isAdmin, editUserAdmin);
router.delete('/users/deleteUserAdmin/:id', isAuthenticated, isAdmin, deleteUserAdmin);
//Acciones de todos los usuarios  

router.get('/users/myProfile', isAuthenticated, myProfile);
router.get('/users/changePasswd', isAuthenticated, changePasswdForm);
router.post('/users/changePasswd', isAuthenticated, changePasswd);
router.get('/users/forgotPasswdForm', forgotPasswordForm);
router.post('/users/forgotPassword', forgotPassword)  

module.exports = router;

```

Figura 17. Enrutador users.routes.js.

Figura 18. Controlador users.controller.js.

Las peticiones y controladores usados en la sección de ingreso de la plataforma son adaptaciones leves de la documentación entregada por el módulo passport-local. La secuencia usada es: renderizado seguido de un proceso. Este apartado es el único sin los helpers de auth.js hasta llegar a la ruta de home.

Una de las principales funciones del admin de la plataforma es la administración de los usuarios que dan uso a esta, ExpressJS nos permite un CRUD para esto. Toda esta sección se encuentra la validación isAdmin para dar una comprobación del role de quien ejecuta estas rutas. Las rutas que no poseen estas comprobaciones son el renderizado del primer administrador de la plataforma se encuentra en la ruta con petición GET "/users/createAdminForm" y la creación mediante la petición POST de la ruta "/users/createAdmin". Si se desea crear más administradores en la plataforma es permitido en la interfaz de creación de usuarios.

Las "Acciones de todos los usuarios" de la plataforma se basan en ver su propio perfil y el cambio de su contraseña. En el caso de pérdida de contraseña, se envía un correo mediante NodeMailer (figura 19).

```

usersCtrl.forgotPassword = async (req, res) => {
    const { email } = req.body;
    const user = await User.findOne({email}).lean();
    if(user != null){
        const {name, lastname, sec_lastname} = user;
        var passwd = short.generate();
        let password = await user.encryptPassword(passwd);
        const id = user._id;
        await User.findByIdAndUpdate(id, { password });
        const message = "Su contraseña ha sido cambiada."
        contentHTML =
            <h1>Módulo CAP</h1>
            <h4>Producción intelectual de docentes</h4>
            <ul>
                <li>Usuario: ${name} ${lastname} ${sec_lastname}</li>
                <li>Email: ${email}</li>
                <li>Nueva contraseña: ${passwd}</li>
            </ul>
        <p>${message}</p>
        let transporter = nodemailer.createTransport({
        });
        let info = await transporter.sendMail({
            from: '"Plataforma PSA - GTST-Usco" <gtst@usco.edu.co>',
            to: email,
            subject: 'Recuperación de contraseña',
            html: contentHTML
        })
    }
    req.flash('success_msg', 'Por favor revisa tu correo electrónico!!!');
    res.redirect('/users/signInForm')
}

```

Figura 19. Envío de correo para recuperar contraseña.

El transportador de nodemailer se configura según la documentación y el correo habilitado con las credenciales para uso de aplicaciones por terceros.

3. PUBLICATIONS

Todo el proceso que abarca la plataforma está basado en la secuencia de una solicitud por productividad académica por parte del docente hasta su aprobación por parte del Comité de Asignación de Puntajes (CAP). En la figura 20 se observa un resumen del proceso de la plataforma el cual consiste primeramente en la creación de la solicitud de productividad académica con los datos correspondientes a esta. Posteriormente, es revisada por el funcionario encargado de la plataforma haciendo una validación manual teniendo acceso a toda la información suministrada por el docente, dictaminando así un rechazo, corrección o aceptación de estos datos, junto a una observación si es necesaria. Esta decisión está basada en las experiencias y opiniones brindadas en las pruebas con los usuarios finales de la plataforma.

El rechazo es motivo de negación total respecto a la solicitud del docente, en cambio una corrección habilita la edición de esta en el listado de solicitudes de la interfaz del docente enviando una notificación al correo.

Al poseer una solicitud válida para la presentación ante el Comité de Asignación y Reconocimiento de Puntaje (CAP) esta se someterá a la revisión por parte de este comité que decide la negación o reconocimiento del puntaje y el tipo de bonificación si es otorgada. Este proceso finaliza con un correo enviado al docente con la decisión tomada por el comité.

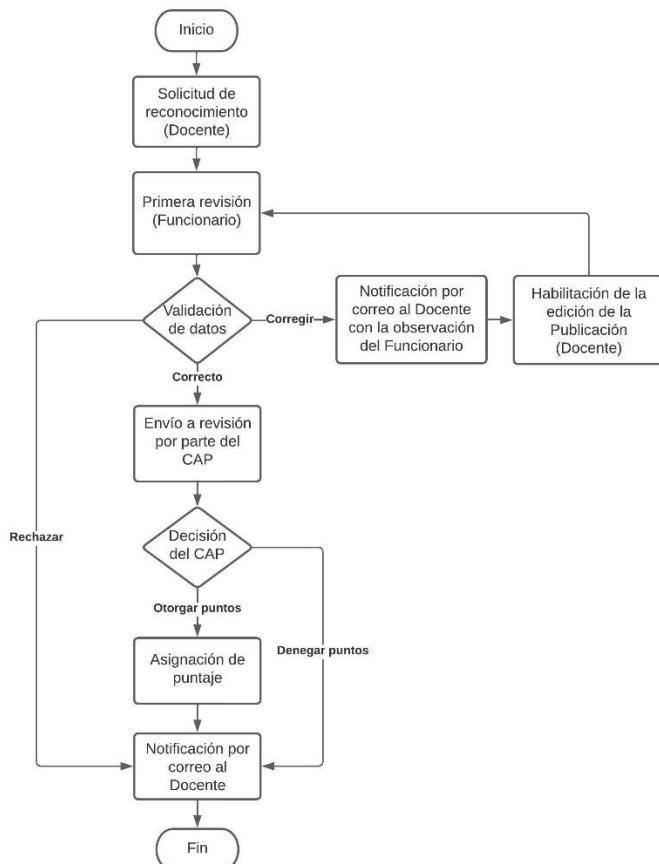


Figura 20. Diagrama de flujo del proceso principal de la plataforma

El archivopublications.routes.js (figura 21) está seccionado para cada uno de los usuarios de la plataforma.

```

//Docente
router.get('/publications/add',isAuthenticated, isDocente, renderAddPublication);
router.post('/publications/add',isAuthenticated, isDocente, AddPublication);
router.post('/publications/verification',isAuthenticated, isDocente, fileVerification);
router.get('/publications/myPublications',isAuthenticated, isDocente, renderMyPublications);
router.delete('/publications/delete/:id',isAuthenticated, isDocente, deleteMyPublication);
router.get('/publications/time',isAuthenticated, isDocente, timeVerification);
router.get('/publications/edit/:id', isAuthenticated, isDocente, renderEditarPublicacion);
router.post('/publications/edit', isAuthenticated, isDocente, editarPublicacion);
//Funcionario
router.get('/publications/audit/fn',isAuthenticated, isFuncionario, renderAuditFn);
router.get('/publications/audit/fn/:id', isAuthenticated, isFuncionario, renderAuditFnId);
router.get('/publications/audit/cap',isAuthenticated, isFuncionario, renderAuditCAP);
router.get('/publications/audit/cap/:id',isAuthenticated, isFuncionario, renderAuditCAPId);
router.post('/publications/reviewed/cap',isAuthenticated, isFuncionario, revisionCAP);
router.get('/publications/search',isAuthenticated, isFuncionario, renderSearchPublication);
router.post('/publications/search',isAuthenticated, isFuncionario, SearchPublication);
router.get('/publications/request/:id', isAuthenticated, isFuncionario, renderRequest);
router.get('/isbn/check', isAuthenticated, isFuncionario, checkISBN)
router.get('/issn/check', isAuthenticated, isFuncionario, checkISSN)
router.post('/publications/reviewed',isAuthenticated, isFuncionario, primeraRevision);
router.get('/file/download/:id', isAuthenticated, dowloadFile);
//Admin
router.get('/load/issn', isAuthenticated, isAdmin, renderLoadISSN)
router.post('/load/issn', isAuthenticated, isAdmin, loadISSN)
router.get('/view/issn', isAuthenticated, isAdmin, renderISSN)
router.get('/load/isbn', isAuthenticated, isAdmin, renderLoadISBN)
router.post('/load/isbn', isAuthenticated, isAdmin, loadISBN)
router.get('/view/isbn', isAuthenticated, isAdmin, renderISBN)

```

Figura 21. Enrutador publications.router.js.

La primera acción en este proceso está descrita con la creación de una publicación ruta “/publications/add” con el controlador “AddPublications”. La creación se hace mediante una petición “AJAX” en la función “fileLimit” en el archivo functions.js con el fin de generar una alerta antes de cargar el archivo con errores. El código se entrega sin límite con el fin de realizar posteriormente en esta sección restricciones de tamaño o tipo de los archivos. El proceso de esta función está descrito en la figura 22.

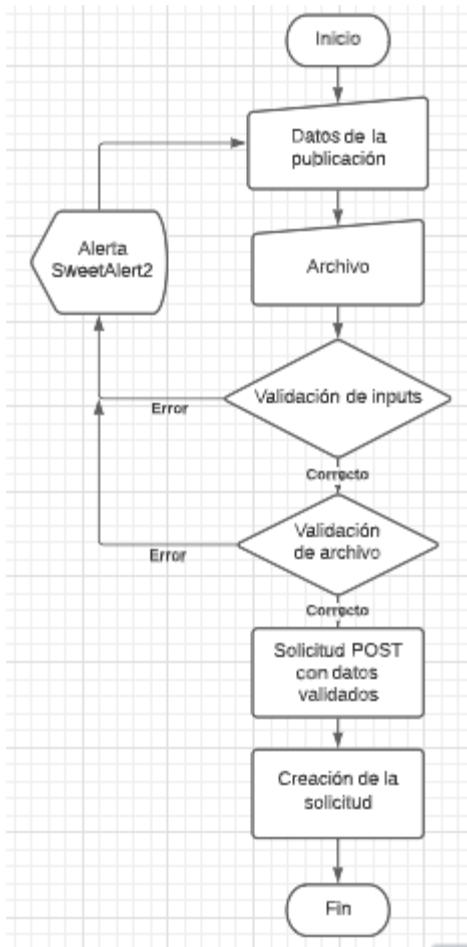


Figura 21. Proceso de creación de una solicitud.

Las vistas del *Docente* se dan mediante peticiones “get” en las vistas de todas las solicitudes que haya creado, además de una vista individual a cada una de estas. La edición de una solicitud se habilitará cuando el funcionario indique una corrección a la solicitud del docente. El renderizado de esta solicitud se adaptará a la modalidad mediante los condicionales de “express-handlebars”.

La eliminación de una solicitud se habilita en los próximos cinco minutos posteriores a la creación de la solicitud. Para ello se usa un condicional comparando fechas con el formato correspondiente. Al ser una plataforma para la Universidad Surcolombiana, el GMT será -5, esto se tiene en cuenta en el momento de la creación de las fechas y explica el uso de una variable “fecha_solicitud” en vez de la variable generada automáticamente por la base de datos “createdAt”.

Los procesos del “Funcionario” se basan en dos revisiones. La primera revisión de los datos de la solicitud se hace en un formulario con la información en “inputs” deshabilitados, una casilla para las observaciones y tres opciones a seguir. Este formulario se renderiza mediante una ruta con la variable “id” para su búsqueda en la base de datos.

El funcionario podrá comprobar en la base de datos de los seriales ISSN e ISBN si la modalidad aplica a ellos mediante un ícono al lado de estos seriales que ejecuta la función “checkISSN” (figura 22) ó “checkISBN”, similar a la anterior, con una petición AJAX y una alerta con el resultado con SweetAlert2.

```
function checkISSN(data){  
    var id = data.attributes[0].value;  
    $.ajax({  
        type: "get",  
        url: "/issn/check",  
        data: {id},  
        success: function(r){  
            if(r.validation){  
                Swal.fire({ ...  
                });  
            }else{  
                Swal.fire({ ...  
                });  
            }  
        },  
        error: function (e) {  
            Swal.fire({ ...  
            });  
            console.log("error: ", e);  
        }  
    });  
}
```

Figura 23. Función checkISSN del archivo functions.js.

La selección de las opciones “aceptar”, “rechazar” y “editar” se hace mediante la función “accept”, que cambia el valor en un “input” oculto dependiendo del botón y hace “submit” al formulario.

La revisión del CAP hecha por el funcionario cuenta con un formulario más sencillo y con funcionalidades similares. Según sea el “checkbox” seleccionado se ejecuta la función “otorgarPuntos” (figura 24) ó “denegarPuntos”, dinamizando el formulario con la aparición de nuevas casillas o eliminándolas con jQuery.

```

function otorgarPuntos(){
    $('#rowOtorgados').remove()
    let html = '<div class="row" id="rowOtorgados">
        <div class="col-4">
            <label>Tipo de Puntaje</label>
            <select class="form-select" name="tipo_puntaje" aria-label="Default select example" required>
                <option value="" disabled selected>---</option>
                <option value="1">Bonificación por productividad académica</option>
                <option value="2">Reconocimiento de puntos salariales</option>
            </select>
        </div>
        <div class="col-2 mb-2">
            <label>Puntaje otorgado</label>
            <input type="number" min="0" class="form-control w-50" value="" name="puntaje" required>
        </div>
    </div>
    $('#denegar').after(html)
}

```

Figura 24. Función otorgarPuntos del archivo functions.js.

La pestaña para buscar solicitudes se ejecuta con una petición get mediante AJAX, realizando una búsqueda rápida con los estados de las solicitudes y las fechas de solicitud, listándolas en una tabla.

El administrador tiene la facultad de cargar seriales ISSN e ISBN para agilizar el proceso de comprobación de estos en la primera revisión del funcionario. Los datos deben estar dispuestos en los formatos de las figuras 25 y 26 en un archivo .xlsx.

A	B	C	D	E	F	G	H
No.	TÍTULO	ISSN IMPRESO	ISSN ELECTRÓNICO	ISSN L	INSTITUCIONES EDITORAS	CATEGORÍA	VIGENCIA
1							
2	1						
3	2						
4	3						
5	4						
6	5						
7	6						
8	7						
9	8						
10	9						
11	10						
12	11						
13	12						
14	13						
15	14						
16	15						
17	16						
18	17						
19	18						
20	19						
21	20						
	21						

Figura 25. Formato .xlsx para cargar ISSN.

A No.	B TÍTULO	C IMPRESO	D ELECTRÓNICO	E EDITORIAL	F AÑO DE PUBLICACIÓN
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					

Figura 26. Formato .xlsx para cargar ISBN.

Los módulos de NodeJS utilizados son Multer y XLSX. La figura 27 nos describe el proceso usado para cargar estos datos.

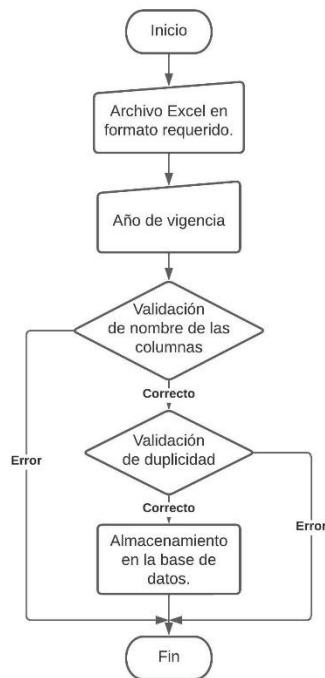


Figura 27. Diagrama de flujo del proceso para cargar ISSN o ISBN.

Multer permite cargar el archivo en la plataforma para ser leído por XLSX y entregarnos una variable con los datos en formato JSON que se recorre mediante un ciclo “for”.

4. REPORTS

En la interfaz del funcionario se encuentra la opción de generar un informe que consta del proceso descrito en la figura 28. Es de forma dinámica ya que tiene tres apartados permitiendo estar o no activos o combinados:

1. Estado y Modalidad: Estos son filtros para la generación de informe que constan del estado en que se encuentran las solicitudes, así como la modalidad académica de estas. Por defecto, todas las casillas de estado se encuentran seleccionadas y el selector de modalidades en la opción “Todas las modalidades académicas”.
2. Año, trimestre y fecha: Esta sección tiene como finalidad dar informes periódicos basándose en estas tres opciones. Si el usuario selecciona un año y trimestre las fechas de solicitud y publicación quedarán deshabilitadas, si seleccionamos algunas de las fechas las demás casillas de esta sección quedarán deshabilitadas, ya que la búsqueda se acotará en un único rango de fechas.
3. Docente, facultad y programa: Este apartado permite seleccionar un docente en específico, un programa o una facultad de la cual se requiera el informe; A su vez posee un buscador por el número de cédula del docente. Esta búsqueda seleccionará la facultad y el programa del docente, si no se selecciona algunas de estas opciones tomará por defecto todas las posibilidades de este selector. La jerarquía para hacer la búsqueda en esta sección es la siguiente: Docente > Programa > Facultad.

Cabe recalcar que si no se encuentra desplegada alguna de estas tres pestañas o ninguna de ellas, el informe se generará con la mayor cantidad de opciones posibles, ya que si no se encuentran seleccionadas no se aplicará el filtro de esa sección.

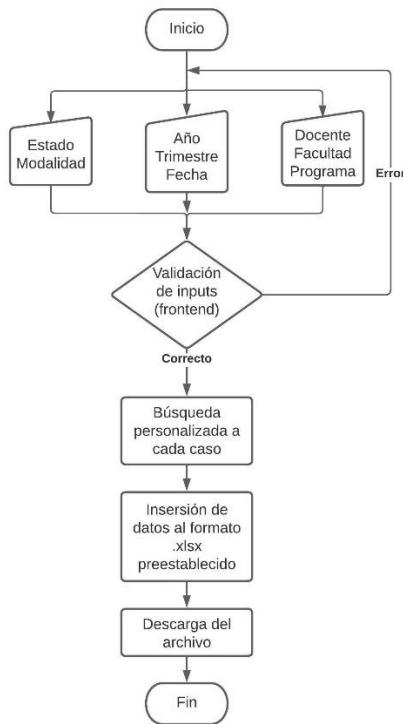


Figura 28. Diagrama de flujo de la generación de informe.

El proceso para dar búsqueda se hace mediante la función “Object.assign” de Javascript a la variable “data”. Se secciona en los tres apartados mencionados anteriormente con la identificación de las variables “switch_1”, “switch_2” y “switch_3” por inputs ocultos que retornan 1 si está activo o 0 inactivo. Así con un condicional sepáramos cada una de las propiedades para asignarlas a “data” y no incurrir en dejar estas con valor *null*. Al final de estos tres condicionales se hace la búsqueda con la variable “data” y trabajar con el JSON retornado en la variable “publications”.

Con el módulo XLSX leemos el archivo “informe.xlsx” de la carpeta xlsx para obtener las casillas en las cuales imprimir los valores de las publicaciones y asignar estos valores en el archivo duplicado “informe2.xlsx”. Así siempre se obtiene un formato en blanco (informe.xlsx) y se descarga el formato diligenciado por ExcelJS (informe2.xlsx).

La inserción de datos mediante el módulo ExcelJS se basa en la documentación de este y la descarga en la función “then” usando el módulo “filesystem” estableciendo los “headers” finalizando con un “res.send”.

```
const Publication = require('../models/publications')
const User = require('../models/User')
const Autor = require('../models/autor')
const moment = require('moment');

reportsCtrl.renderGenerarInforme = (req,res) =>{ ... }

reportsCtrl.searchDocentes = async (req,res) =>{ ... }

reportsCtrl.searchDocenteCC = async (req,res)=>{ ... }

reportsCtrl.GenerarInforme = async (req,res) =>{ ... }

function switchToBoolean(swtch){ ... }
```

MÓDULO CAP



UNIVERSIDAD
SURCOLOMBIANA

MANUAL DE USUARIO

ADMINISTRADOR
DOCENTE
FUNCIONARIO

Freddy Leonardo Cuenca Beltrán

Plataforma web para la administración de la producción intelectual de los docentes de la Universidad Surcolombiana.

TABLA DE CONTENIDO

DESCRIPCIÓN.....	3
1. Login	4
ADMINISTRADOR.....	3
1. Primer usuario administrador de la plataforma.....	5
2. Agregar usuario.....	5
3. Seriales ISSN e ISBN.....	6
DOCENTE	8
1. Creación de una solicitud.....	8
2. Administración de las solicitudes	9
FUNCIONARIO.....	10
1. Listado de solicitudes pendientes por revisión.....	10
2. Interfaz para la primera revisión.....	10
3. Revisión CAP.....	10
4. Interfaz de generación de informe.....	10

DESCRIPCIÓN

El presente manual de usuario está dividido en 3 secciones, la primera sección corresponde a: **ADMIN:** Administrador de la plataforma. Tiene la responsabilidad de la creación, edición y eliminación de los diferentes usuarios de la plataforma. También tiene la facultad de cargar los seriales ISBN e ISSN. La segunda sección es el **DOCENTE:** Autor de la propiedad académica. Dicha propiedad académica se reconoce mediante una solicitud creada por este usuario, que se revisa y evalúa por el Comité de Asignación de Puntaje (CAP). Y, por último, el **FUNCIONARIO:** Principal actuador del proceso de reconocimiento de una productividad académica. Sus funciones son revisar los datos de las solicitudes creadas por los docentes y diligenciar la decisión emitida por el CAP. Finalmente se genera un informe con los datos de las solicitudes agrupados según las necesidades de los funcionarios expertos en este proceso

1. LOGIN

Todo usuario de la plataforma debe ingresar las credenciales dadas por el administrador en el formulario login para poder ingresar y estar autenticado. (Figura 1).

The screenshot shows the login interface for the 'Módulo CAP' (Producción Intelectual de los Docentes) platform. The page has a dark red header with the text 'Módulo CAP' and a small icon. Below the header, there's a navigation menu with 'Inicio' and a logo for 'UNIVERSIDAD SURCOLOMBIANA'. The main content area is titled 'Módulo CAP' and 'Producción Intelectual de los Docentes', with a subtitle 'Plataforma para la administración de la Producción Intelectual de los Docentes en la Universidad Surcolombiana'. It features two input fields for 'Correo electrónico' and 'Contraseña', a 'Entrar' button, and a link '»> Olvidé mi contraseña <<'. At the bottom, there's a logo for 'GETI' and the text 'Grupo de Investigación en electrónica, telecomunicaciones e informática - GETI' and '© Universidad Surcolombiana'.

Figura 1. Formulario Login.

ADMINISTRADOR

1. Primer usuario administrador de la plataforma

La base de datos no posee un administrador, por lo tanto, la ruta “<https://<dominio>/users/createAdminForm>” permite la creación del primer usuario Admin. A esta no se podrá acceder luego de la creación del primer administrador de la plataforma por lo que si existe se redirecciona a la vista de login. (Figura 2).

The screenshot shows a web browser window with the URL localhost:3000/users/createAdminForm. The page has a dark header with the text "Módulo CAP" and "Iniciar sesión". On the left, there's a sidebar with the "UNIVERSIDAD SURCOLOMBIANA" logo and the word "Inicio". The main content area is titled "Agregar el primer administrador". It contains several input fields:

- Información personal:** Includes dropdowns for "Elija un tipo de identificación" and "No. de identificación", and text inputs for "Nombres", "Primer apellido", and "Segundo apellido".
- Datos de localización:** Includes dropdowns for "Teléfono", "Dirección", "Departamento", and "Municipio".
- Información de sesión:** Includes dropdowns for "Administrador de la plataforma" and "Confirmar contraseña", and a text input for "Contraseña".

At the bottom right of the form is a red "Crear" button.

Figura 2. Formulario para el primer administrador de la plataforma.

2. AGREGAR USUARIO

En la creación de un usuario (figura 3), se encuentran cada uno de los parámetros requeridos para la administración de estos, a través, de toda la plataforma. En el caso de los docentes, se despliegan unas opciones extra necesarias para el informe final.

Módulo CAP

Freddy Leonardo Cuenca Beltrán

Cerrar sesión

Información personal
-- Elija un tipo de identificación -- Nombres Primer apellido Segundo apellido

Datos de localización
Teléfono Dirección
-- Departamento -- Municipio

Información de sesión
Docente Email
Seleccione una Facultad Selecione un Programa

Vinculación
 Tiempo completo planta
 Medio tiempo planta

Pertenece a un grupo de investigación?
 Si
 No

Contraseña Confirmar contraseña

Agregar

Logo: Universidad Surcolombiana
Grupo de investigación en electrónica, telecomunicaciones e informática - GETI
© Universidad Surcolombiana

<https://freddylicuena.com/users/createUserForm#pageSubmenu>

Figura 3. Formulario para agregar un usuario a la plataforma.

3. SERIALES ISSN E ISBN

El administrador de la plataforma cuenta con la interfaz para cargar los seriales ISSN e ISBN utilizados en la primera revisión realizada por el funcionario para comprobar la existencia de estos. Los seriales deberán estar dispuestos en un archivo .xlsx descritos en los formatos de las figuras 4 y 5, donde lo esencial de estos es tener correctamente escritos los títulos de las columnas en la fila 1.

A Nº.	B TÍTULO	C ISSN IMPRESO	D ISSN ELECTRÓNICO	E ISSN L	F INSTITUCIONES EDITORAS	G CATEGORÍA	H VIGENCIA
1							
2	1						
3	2						
4	3						
5	4						
6	5						
7	6						
8	7						
9	8						
10	9						
11	10						
12	11						
13	12						
14	13						
15	14						
16	15						
17	16						
18	17						
19	18						
20	19						
21	20						
22							

Figura 4. Formato para cargar masivamente seriales ISSN.

	A No.	B TÍTULO	C IMPRESO	D ELECTRÓNICO	E EDITORIAL	F AÑO DE PUBLICACIÓN
1						
2						
3						
4						
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						

Figura 5. Formato para cargar masivamente seriales ISBN.

DOCENTE

1. CREACIÓN DE UNA SOLICITUD

El primer paso en el proceso de aceptación y administración de la productividad académica de un docente es la creación de una solicitud de productividad académica por parte de este usuario. El formulario de la figura 6 se adapta fácilmente a cada una de las modalidades académicas mostrando los apartados de cada una de ellas. Las casillas desplegadas deberán ser correctamente diligenciadas ya que este aplicativo web hace validaciones a los datos suministrados por el usuario.

The screenshot shows the 'Solicitud de Reconocimiento' (Recognition Request) form. The form fields include:

- Título del Material: Nombre del Artículo (text input), Fecha de Publicación (date input mm/dd/aaaa), Tipo de Artículo (dropdown menu).
- Artículo de Revista (dropdown menu), Nombre de la Revista (text input), Categoría de la revista en Minciencias al momento de la publicación (dropdown menu).
- Dirección URL (text input), Seleccionar archivo (file input), Ninguno archivo selec. (radio button).
- Tiempo requerido para la publicación del artículo en la revista (text input), Cantidad en meses (text input).
- Fecha de recepción o sumisión del artículo a la revista (date input mm/dd/aaaa).
- ISSN (text input).
- Para la publicación del artículo, se invirtieron recursos económicos de la Universidad (radio buttons Si / No).
- Nombre del proyecto de Investigación del cual se genera el material, si aplica (text input).
- El material presentado fue tenido en cuenta para cambio de categoría en el escalafón docente (radio buttons Si / No).
- Número de Autores (dropdown menu showing 1).
- Enviar (button).

Figura 6. Formulario para la creación de una solicitud de productividad académica.

2. ADMINISTRACIÓN DE LAS SOLICITUDES.

Una vez diligenciada y enviada la solicitud, el docente tendrá la opción de eliminar esta hasta 5 min luego de su creación. Para ello cuenta con una tabla dando click a “Ver mis solicitudes” (figura 7) donde se listan todas las solicitudes hechas en orden ascendente a la fecha de creación.

The screenshot shows a web application interface for managing research requests. On the left, there's a sidebar with a red background containing links: 'Modulo CAP', 'Inicio', 'Solicitudes', 'Crear Solicitud', 'Ver mis Solicitudes', 'Perfil', 'Ver perfil', and 'Cambiar mi contraseña'. Below these is the 'SURCOLOMBIANA' logo. The main content area has a white header with 'Solicitudes'. Below it is a table with the following columns: #, Nombre, Modalidad, URL, Fecha de Solicitud, Fecha de publicación, Estado, Descargar, and Eliminar. There are five rows of data:

#	Nombre	Modalidad	URL	Fecha de Solicitud	Fecha de publicación	Estado	Descargar	Eliminar
005	Ejemplo_Estudiodpostdoctoral	estudios postdoctoriales	#	15/01/2023	09/07/2022	Pendiente por revisión	#	#
004	Ejemplo_Producciónsoftware	producción de software	#	15/01/2023	10/11/2022	Pendiente por revisión	#	#
003	Ejemplo_Revista4	Artículo de Revista	#	15/01/2023	16/12/2022	Pendiente por revisión	#	#
002	Ejemplo_Revista3	Artículo de Revista	#	15/01/2023	08/09/2022	Pendiente por revisión	#	#
001	Ejemplo_Libro2	libro	#	15/01/2023	14/10/2022	Pendiente por revisión	#	#

At the bottom right of the table, there's a link 'búsqueda en tabla' and a small logo. At the very bottom right of the page, there's a 'Cerrar sesión' button.

Figura 7. Tabla de solicitudes realizadas por el docente.

FUNCIONARIO

1. LISTADO DE SOLICITUDES PENDIENTES POR REVISIÓN.

El funcionario de la plataforma tiene como responsabilidad, realizar una revisión manual de todos los datos proporcionado por el docente en el momento de la creación de la solicitud, por lo tanto, en su interfaz se encuentra un listado de las solicitudes pendientes por revisión (Figura 8).

#	Docente	Facultad	Programa	Nombre de Publicación	URL	Fecha de Solicitud	Descargar	Revisar
1	pruebaDocente33 Ejemplo 33 Prueba 33	Ciencias Exactas y Naturales	Matemática Aplicada	Ejemplo_Revista1	⊕	15/01/2023	▲	■
2	pruebaDocente33 Ejemplo 33 Prueba 33	Ciencias Exactas y Naturales	Matemática Aplicada	Ejemplo_Revista2	⊕	15/01/2023	▲	■
3	pruebaDocente33 Ejemplo 33 Prueba 33	Ciencias Exactas y Naturales	Matemática Aplicada	Ejemplo_Libro	⊕	15/01/2023	▲	■
4	pruebaDocente33 Ejemplo 33 Prueba 33	Ciencias Exactas y Naturales	Matemática Aplicada	Ejemplo_Patente1	⊕	15/01/2023	▲	■
5	pruebaDocente33 Ejemplo 33 Prueba 33	Ciencias Exactas y Naturales	Matemática Aplicada	Ejemplo_Ponencia1	⊕	15/01/2023	▲	■
6	Docente 2 Ejemplo 2 Prueba 2	Economía y Administración	Contaduría Pública	Ejemplo_Libr2	⊕	15/01/2023	▲	■
7	Docente 2 Ejemplo 2 Prueba 2	Economía y Administración	Contaduría Pública	Ejemplo_Revista3	⊕	15/01/2023	▲	■
8	Docente 2 Ejemplo 2 Prueba 2	Economía y Administración	Contaduría Pública	Ejemplo_Revista4	⊕	15/01/2023	▲	■
9	Docente 2 Ejemplo 2 Prueba 2	Economía y Administración	Contaduría Pública	Ejemplo_Producciónsoftware	⊕	15/01/2023	▲	■
10	Docente 2 Ejemplo 2 Prueba 2	Economía y Administración	Contaduría Pública	Ejemplo_Estudiodpostdoctoral	⊕	15/01/2023	▲	■
11	Docente1 Ejemplo Prueba	Ciencias Exactas y Naturales	Física	Ejemplo_dreccióntesis	⊕	15/01/2023	▲	■
12	Docente1 Ejemplo Prueba	Ciencias Exactas y Naturales	Física	Ejemplo_Produccióntécnica	⊕	15/01/2023	▲	■

Figura 8. Listado de solicitudes pendientes por primera revisión.

2. INTERFAZ PARA LA PRIMERA REVISIÓN.

Los formularios para la creación de solicitudes por parte de los docentes son adaptados a la modalidad académica seleccionada, de igual manera, en la primera revisión por parte del funcionario las casillas de cada una de las propiedades de la solicitud seleccionada se ajustarán dependiendo de la modalidad seleccionada.

Las opciones “Aceptar”, “Rechazar” o “Editar”, así como el campo para las observaciones si lo requiere, son seleccionadas o diligenciadas por el funcionario, dando paso al siguiente estado de la solicitud. En la figura 9 se observa la interfaz de lo anteriormente dicho.

Figura 9. Interfaz de la primera revisión.

3. REVISIÓN CAP.

El Comité de Asignación de Puntaje somete a evaluación las solicitudes de producción académica que hayan sido aceptadas por el funcionario. Al obtener la respuesta, el funcionario diligencia el formulario “Revisión CAP”, el cual despliega un seleccionador para el tipo de bonificación y una casilla para el puntaje si la opción seleccionada es “Otorgar puntos” (figura 10). Por último, se ingresa el concepto del CAP por el cual se otorgan o no puntos a la solicitud.

Figura 10. Formulario de revisión CAP.

4. INTERFAZ DE GENERACIÓN DE INFORME.

El funcionario tiene la capacidad de generar un informe con los datos filtrados según se requiera. Cada uno de los apartados de esta interfaz (Figura 11) posee una validación en el frontend según se seleccionen las opciones deseadas. Por defecto la sección de “Estado y Modalidad” tendrá seleccionadas todas las opciones de estado y todas las modalidades. Para el apartado de “Año, Trimestre y Fecha” la validación está dividida entre Año-Trimestre, Fecha de Solicitud y Fecha de Publicación, habilitando únicamente la opción seleccionada y deshabilitando las otras. Por último, la validación hecha en “Docente, facultad, programa” tiene una jerarquía donde la selección de Docente es el principal filtro, seguido del Programa y por último la Facultad. Cabe aclarar que cada uno de los apartados genera un criterio en el filtro independiente entre sí.

La captura de pantalla muestra la interfaz de usuario para la generación de informes. En la parte superior izquierda, hay un menú vertical con las siguientes opciones: Módulo CAP, Inicio, Solicitudes, Informes, Generar Informe, Perfil. Dentro de este menú, se encuentra el logo de la Universidad Surcolombiana. La parte central de la pantalla es un formulario titulado "Búsqueda de Solicitudes". El formulario se divide en tres secciones principales:

- Estado y Modalidad:** Una sección que incluye "Estado de las Solicitudes" (con opciones como Pendiente por revisión, Revisado, Editar, Rechazado, No aprobado por CAP y Aprobado), "Modalidad Académica" (que muestra "Todas las modalidades") y un botón para "Cerrar sesión".
- Año, Trimestre, Fecha:** Una sección que permite filtrar por "Año" (selecciónable), "Trimestre" (selecciónable), "Fecha de Publicación (Inicio)" y "Fecha de Publicación (Fin)" (ambas con formato dd/mm/aaaa), y "Fecha de Solicitud (Inicio)" y "Fecha de Solicitud (Fin)" (ambas con formato dd/mm/aaaa).
- Docente, Facultad, Programa:** Una sección que incluye "Todas las Facultades" (selecciónable), "Todos los Programas" (selecciónable), un campo para "Ejemplo Docente" (que muestra "Buscar Docente por Cédula") y un campo para "Cédula" con un icono de lupa.

Figura 11. Interfaz de generación de informe.