

Diseño e implementación de una herramienta tecnológica para el mejoramiento en la gestión de reuniones del Instituto de Proyección Regional y Educación a Distancia, IPRED, de la Universidad Industrial de Santander

Andrés Felipe Camacho Velandia

Trabajo de Grado para Optar el título de Ingeniero de Sistemas

Director

David Alejandro Miranda Mercado, Ph.D

Profesor Titular de la Escuela de Física

Codirectora

María Alejandra Acevedo Valencia

Ingeniera de Sistemas, Profesional del IPRED

Universidad Industrial de Santander

Facultad de Fisicomecánicas

Escuela de Ingeniería de Sistemas

Bucaramanga

2019

Contenido

Introducción	11
1. Objetivos	13
1.1 Objetivo General	13
1.2 Objetivos Específicos	13
2. Marco Teórico	14
2.1 Tecnologías de la Información y Comunicación	14
2.2 Gobierno Electrónico	15
2.3 Gobierno En Línea	17
2.4 Instituto de Proyección Regional y Educación a Distancia	18
2.5 Herramientas de desarrollo	19
2.5.1 MongoDB	20
2.5.2 Node.JS	21
2.5.3 React	21
2.6 REST	22
2.7 SCRUM	23
3. Diseño de la aplicación web	26
4. Estructura de datos e interfaz de usuario	34
4.1 Estructura de datos	34
4.2 Diseño de interfaz	38
5. Descripción de los módulos diseñados	43
5.1 Módulo de Back-End	43
5.1.1 Módulo de autenticación	44
5.1.2 Módulo de Almacenamiento	45
5.1.3 Módulo de Datos	45
5.2 Módulo de Front-End	49
5.2.1 Módulo de Cliente REST	51
5.2.2 Módulo de Cliente de Autenticación	51
5.2.3 Módulo de Almacenamiento	52

5.2.4 Módulo de Ingreso y Consulta	52
5.2.5 Módulo de Interfaz de Usuario	54
5.2.6 Módulo de Recursos	54
5.2.7 Módulo de exportar	55
5.2.8 Módulo de Lenguajes	55
6. Algoritmos implementados	56
6.1 Algoritmo de Autenticación y Obtención de Token.	56
6.2 Algoritmo de petición REST	58
6.3 Algoritmo de Exportar Acta	59
6.4 Algoritmo de búsqueda de temas por palabra clave	60
7. Evaluación y pruebas de la aplicación web	63
7.1 Usabilidad	65
7.1.1 Comprensibilidad	65
7.1.2 Documentación	66
7.1.3 Capacidad de compilación.	67
7.1.4 Capacidad de instalación	69
7.1.5 Curva de aprendizaje	69
7.2 Sostenibilidad y mantenibilidad	70
7.2.1 Derechos de autor	70
7.2.2 Accesibilidad	71
7.2.3 Portabilidad	72
7.2.4 Capacidad de soporte	73
7.2.5. Analizabilidad	74
8. Conclusiones	76
Referencias Bibliográficas	78
Apéndices	80

Lista de Figuras

Figura 1. Story Mapping de rol de secretaria de SIGER.	27
Figura 2. Story Mapping de roles de coordinador y consulta de SIGER.....	28
Figura 3. Diseño de alto nivel para aplicación web SIGER.	33
Figura 4. Modelo de datos para Acta.	35
Figura 5. Modelo de datos para actividad, tema y sus respectivas relaciones.	37
Figura 6. Vista de creación de Acta en SIGER.....	39
Figura 7. Vista de creación de acta, pestaña de actividades en SIGER.	39
Figura 8. Vista de lista de actas creadas en SIGER.	40
Figura 9. Vista de edición de Acta.....	41
Figura 10. Vista de SIGER. A la izquierda menú de recursos. A la derecha vista de edición de acta.	42
Figura 11. Diagrama de módulos del Back-end.....	44
Figura 12. Definición de modelo de datos de acta con archivo JSON.	46
Figura 13. Diagrama de módulos del front-end.	50
Figura 14. Vista de consulta en SIGER.	53
Figura 15. Vista de ingreso en SIGER.....	53
Figura 16. Componente de agregar temas.....	54
Figura 17. Diagrama de algoritmo de autenticación y obtención de token.	56
Figura 18. Diagrama de algoritmo de petición REST.....	58
Figura 19. Diagrama de algoritmo de exportar acta.	59
Figura 20. Diagrama de algoritmo de búsqueda de temas por palabra clave.	60
Figura 21. Resultado de algoritmo de búsqueda por palabra clave usando “Maria”.	62

Lista de Tablas

Tabla 1.	30
<i>Herramientas propuestas para desarrollo de SIGER.</i>	30
Tabla 2.	64
<i>Tabla de criterios de evaluación agrupada.</i>	64
Tabla 3.	65
<i>Tabla de criterios de comprensibilidad.</i>	65
Tabla 4.	66
<i>Tabla de criterios de documentación.</i>	66
Tabla 5.	68
<i>Tabla de criterios de capacidad de compilación.</i>	68
Tabla 6.	69
<i>Tabla de criterios de capacidad de instalación.</i>	69
Tabla 7.	69
<i>Tabla de criterios de curva de aprendizaje.</i>	69
Tabla 8.	71
<i>Tabla de criterios para derechos de autor.</i>	71
Tabla 9.	71
<i>Tabla de criterios para accesibilidad.</i>	71
Tabla 10.	72
<i>Tabla de criterios para portabilidad.</i>	72
Tabla 11.	73
<i>Tabla de criterios para capacidad de soporte.</i>	73
Tabla 12.	74
<i>Tabla de criterios para analizabilidad.</i>	74

Lista de Apéndices

Apéndice A. Descripción del hardware y software requeridos para el despliegue.....	80
Apéndice B. Descripción de la instalación.	83
Apéndice C. Manual de Usuario.....	91

Resumen

TÍTULO: DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA TECNOLÓGICA PARA EL MEJORAMIENTO EN LA GESTIÓN DE REUNIONES DEL INSTITUTO DE PROYECCIÓN REGIONAL Y EDUCACIÓN A DISTANCIA, IPRED, DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER ¹

AUTOR: ANDRÉS FELIPE CAMACHO VELANDIA ²

PALABRAS CLAVE: GOBIERNO ELECTRÓNICO, IPRED, WEB, APLICACIÓN WEB, REACT, MONGODB, AZURE, UBUNTU, NODEJS

DESCRIPCIÓN: El gobierno electrónico es la aplicación de las tecnologías de la información y la comunicación, TIC, a la administración de información de una entidad gubernamental. Las herramientas del gobierno electrónico disminuyen los contratiempos presentados en la gestión de procesos gubernamentales, permitiendo enfocar recursos en otras secciones y contribuyendo a la mejora de los servicios ofrecidos. El Instituto de Proyección Regional y Educación a Distancia, IPRED, es una unidad de la Universidad Industrial de Santander encargada de la formación, investigación y extensión de las sedes regionales. Semanalmente el IPRED recibe una serie de solicitudes que deben ser resueltas, estas peticiones son compiladas en un documento denominado acta, debido a que es un documento compartido y manipulado por múltiples personas sin ningún control, existen problemas en la creación y distribución de actas, que se traduce en demoras en su publicación para consulta. Este proyecto propone e implementa una aplicación llamada SIGER o Sistema de Gestión de Reuniones, que da soporte a la administración de datos generados por los consejos del Instituto de Proyección Regional y Educación a Distancia, con el objetivo de disminuir el tiempo de creación y distribución de información generada por los consejos. La aplicación agrega opciones de búsqueda rápida, creación, exportación de manera intuitiva, también funciona como un repositorio de archivos para consulta. La aplicación es implementada siguiendo los lineamientos dados por el IPRED sobre despliegue en la nube, facilidad de configuración y rápido despliegue. En este proyecto se presenta el diseño, implementación y despliegue de la aplicación para su uso en los consejos del IPRED.

¹ Trabajo de grado

² Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas. Director: David Alejandro Miranda Mercado.

Abstract

TITLE: DESIGN AND IMPLEMENTATION OF A TECHNOLOGICAL TOOL FOR THE IMPROVEMENT IN THE MANAGEMENT OF MEETINGS OF THE INSTITUTE OF REGIONAL PROJECTION AND DISTANCE EDUCATION, IPRED, OF THE UNIVERSIDAD INDUSTRIAL DE SANTANDER ³

AUTHOR: ANDRÉS FELIPE CAMACHO VELANDIA ⁴

KEYWORDS: E-GOVERNMENT, IPRED, WEB, APLICACIÓN WEB, REACT, MONGODB, AZURE, UBUNTU, NODEJS

DESCRIPTION: Electronic government is the application of information and communication technologies, ICT, to the administration of information of a government entity. The tools of electronic government reduce the setbacks presented in the management of government processes, allowing to focus resources in other sections and contributing to the improvement of the services offered. The Institute for Regional Projection and Distance Education, IPRED, is a unit of the Industrial University of Santander in charge of training, research and extension of regional faculties. Weekly, the IPRED receives a series of requests that must be resolved, these requests are compiled into a document called “act”, because it is a document shared and manipulated by multiple people without any control, there are problems in the creation and distribution this documents, which translates into delays in its publication for consultation. This project proposes and implements an application called SIGER or Meeting Management System, which supports the administration of data generated by the councils of the Institute of Regional Projection and Distance Education, with the objective of reducing the creation and distribution time of information generated by the councils. The application adds options for quick search, creation, and export of this information, also works as a file repository for consultation. The application is implemented following the guidelines given by the IPRED on cloud deployment, ease of configuration and rapid deployment. This project presents the design, implementation and deployment of the application for use in the IPRED councils.

³ Bachelor Thesis

⁴ Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas. Director: David Alejandro Miranda Mercado.

Introducción

El Instituto de Proyección Regional y Educación a Distancia, IPRED, es una unidad de la Universidad Industrial de Santander, UIS, que posee la función de promover la formación, investigación y extensión de las sedes regionales y programas a distancia impartidos por la universidad. Para realizar estas funciones posee tres consejos, El Consejo de Sedes, El Consejo de Instituto y El Consejo de Programas. Estos consejos reciben peticiones o propuestas relacionadas con asuntos del IPRED, los Miembros del Consejo evalúan estas propuestas, las compilan en un documento y finalmente las publican en forma de acta. En el proceso de creación de un acta interactúan las secretarías del IPRED y los Miembros del Consejo, y es escrita en medios digitales usando software de ofimática, como Microsoft Word. Aunque algunos softwares de ofimática ofrecen edición colaborativa, en ocasiones produce pérdida de datos y entradas no autorizadas. Lo anterior ocurre por una falta de protocolo de administración de información de actas. Adicionalmente, en la edición y publicación del acta existen una serie de reprocesos que aplazan la publicación del documento firmado. Otro aspecto es la consulta del acta, no existe un repositorio digital que permita buscar un acta específica, ni los temas que se trataron, lo que dificulta la distribución de información entre unidades académico-administrativas.

El gobierno electrónico es la aplicación de las tecnologías de la información y comunicación, TIC, a la administración de la información de una entidad gubernamental, reduciendo la burocracia presentada en los servicios que ofrece [13]. La digitalización de estos procesos promueve una mayor participación y acceso a los ciudadanos y actores del gobierno [13]. El concepto ha tomado

popularidad en el siglo XXI, debido a la proliferación de Internet y dispositivos portátiles, que han permitido que el gobierno electrónico tenga un mayor alcance [13]. Gobierno en Línea es la apuesta del Gobierno Colombiano por promover estas políticas, ya que permiten no solo un cambio estructural en las entidades, sino también permiten la comunicación y distribución de información entre entidades gubernamentales y ciudadanos [7]. En este sentido y teniendo presente la problemática de información de actas del IPRED, este proyecto introduce una aplicación web para la administración y distribución de actas generadas por consejos del IPRED, con el propósito de mejorar la cultura organizacional asociada a los Consejos del Instituto y optimizar la distribución de la información.

Este libro cuenta con 8 capítulos. El primero expone los objetivos generales y específicos. El segundo corresponde al marco teórico, la información necesaria para entender el alcance proyecto. Los siguientes cuatro capítulos, corresponden al diseño de la base de datos, diseño de algoritmos e interfaz y sus respectivas descripciones. El séptimo capítulo muestra la evaluación y pruebas realizadas. El octavo y último capítulo describe las conclusiones de este proyecto.

1. Objetivos

1.1 Objetivo General

Diseñar e implementar una herramienta digital que permita mejorar la gestión de la información de reuniones de los consejos del Instituto de Proyección Regional y Educación a Distancia, IPRED.

1.2 Objetivos Específicos

- Establecer los requisitos del sistema para atender a las necesidades de gestión de información de las reuniones de consejos del IPRED.
- Diseñar una aplicación web cumpliendo con los requisitos establecidos para el sistema.
- Implementar una aplicación web que pueda ser usada en diferentes dispositivos.
- Realizar las pruebas de funcionamiento de la aplicación web para su uso en el IPRED.

2. Marco Teórico

En este proyecto se propone el desarrollo de una aplicación web que permita la administración de la información generada en las reuniones de consejos del IPRED, cumpliendo con la perspectivas de proyección digital del Gobierno Colombiano, y en concordancia con lo descrito en el decreto 2573 de 2014; dicho decreto recomienda a las entidades públicas implementar las políticas del Gobierno Digital, en donde se promueve el uso de las tecnologías de información y comunicación TIC para ofrecer servicios públicos en diferentes niveles mejorando la eficiencia de los procesos y promoviendo la participación activa de todos los actores involucrados en estos procesos. A continuación, se definen algunos conceptos necesarios para comprender el alcance de este proyecto.

2.1 Tecnologías de la Información y Comunicación

Según la UNESCO, las Tecnologías de Información y Comunicación, TIC, son un conjunto diverso de herramientas tecnológicas usadas para la transmisión, almacenamiento, creación e intercambio de información [15]. Las herramientas TIC incluyen programas de computador e internet, y son usadas en diferentes dominios como la medicina, las operaciones bancarias, la ingeniería, el entretenimiento, la educación, entre otros [3]. En países en desarrollo, donde las distancias físicas y la falta de desarrollo e infraestructura presentan un problema para la educación, las TIC ayudan a su distribución, eliminando este tipo de barreras, y también proponiendo un modelo de educación novedoso, que en el caso de la ciencia y la ingeniería resulta provechoso para

el desarrollo de comunidades aisladas [10]. Algunos estudios han demostrado cómo el uso de las TIC ayuda incrementar el nivel socioeconómico de países en desarrollo [9]. El impacto de las TIC en el siglo XXI ha llevado a la creación de nuevos modelos económicos basados en ciencia y tecnología, este tipo de modelos son provechosos pues reducen el impacto medioambiental y proveen alternativas a los modelos económicos basados en recursos naturales [11]. Es por esto que en varios países fueron implementadas políticas para promover el uso de las TIC, en el caso de Colombia, fue creada con este propósito la Ley 1341 del 2009 [1] y con ella la fundación del Ministerio de Tecnologías de Información y Comunicación.

2.2 Gobierno Electrónico

El gobierno electrónico comprende el uso de las TIC aplicadas a la mejora de las políticas públicas y operaciones gubernamentales, con el objetivo de comprometer a los ciudadanos y proveer mejores servicios gubernamentales [1]. En términos más simples, consiste en usar internet y la Word-Wide-Web para entregar información y servicios gubernamentales a sus ciudadanos [5]. Es un campo de estudio multidisciplinario cuyas áreas están cercadas por las ciencias de la computación, ciencias políticas y de administración, sociología y psicología [1]. Existen tres categorías principales de gobierno electrónico, y dos secundarias. Las categorías están nombradas dependiendo de la entidad que genera la información y el actor que la recibe. Las principales son Gobierno a Gobierno (G2G), Gobierno a Ciudadano (G2C), Gobierno a Empresa (G2B) siendo la última útil para compartir información sobre comercio y colaboración con empresas. Las secundarias son Gobierno a Organizaciones de Sociedad Civil (G2SC) útiles para proveer servicios de comunicación y coordinación de esfuerzos después de un desastre. Finalmente se encuentra

Ciudadano a Ciudadano (C2C), que ofrece servicios de discusión en grupos sobre temas relacionados a políticas gubernamentales.

Los primeros estudios de gobierno electrónico nacieron en los años noventa a partir de la combinación de tecnología con teorías de la administración [1]. Pero, no fue hasta la introducción de internet y la proliferación de las computadoras personales que el gobierno electrónico comenzó a tener protagonismo.

Estados Unidos fue pionero en ofrecer sus servicios gubernamentales públicos en la web, con el propósito de facilitar el manejo de información y distribución de bases de datos. De igual manera, el gobierno de los Estados Unidos fue el precursor en introducir e implementar el concepto de gobierno electrónico [5]. URBIS o Sistema de Información Urbano, fue el primer proyecto creado por la Universidad de California en 1973 con el propósito de usar computadores para la implementación de servicios empresariales complejos [5]. En los siguientes años se hacen diferentes esfuerzos para que se use la web para compartir información gubernamental. En 2001, se crea el Acta de Gobierno Electrónico, que propone un modelo para la distribución de aplicaciones web orientadas a la administración de datos gubernamentales. Después de los ataques del 9/11, se agrega un componente de seguridad a las aplicaciones y bases de datos compartidas debido a la histeria por posibles ataques a la información [5].

Actualmente, el gobierno electrónico es promovido en todo el mundo, ya que demuestra que estas herramientas transforman la relación Estado-Sociedad de manera positiva [8], con este propósito las Naciones Unidas hacen anualmente una encuesta de Índice de Gobierno Electrónico

donde clasifica países y en la cual Colombia se encuentra entre los primeros 50 puestos demostrando un alto índice de desarrollo en este ámbito [8].

2.3 Gobierno En Línea

En el artículo 13 de la Constitución Política de Colombia se menciona que los órganos del estado poseen diferentes funciones, pero deben colaborar de manera eficiente y armoniosa para llevar a cabo sus funciones, es por esto y siguiendo la tendencia global de gobierno electrónico que se crea el decreto 2573 de 2014, en donde el MinTic propone lineamientos para implementar el gobierno electrónico en Colombia en un documento llamado Manual Gobierno En Línea. En este decreto también se determina que el Estado y sus diferentes órganos deben promover un acceso eficiente e igualitario para los habitantes del territorio nacional.

Según López-De Castro y García Alonso [7], el Manual Gobierno En Línea “(...) está conformado por cuatro (4) componentes fundamentales que guían la implementación del *e-government* por parte de entidades del orden nacional y territorial, a saber: TIC para servicios, TIC para el gobierno abierto, TIC para la gestión, seguridad y privacidad de la información. A partir de la definición de estos componentes fundamentales, el gobierno establece unas metas y etapas de implementación de acuerdo a la categoría de las entidades territoriales, se crea el índice de Gobierno Electrónico que mide los avances de la estrategia a nivel nacional y se generan herramientas transversales para el acompañamiento de las instituciones, entre ellas un mapa de ruta que establece los trámites y servicios que más necesitan los colombianos, y a la vez una asistencia para construir las páginas web de las entidades (...) la estrategia de Gobierno en Línea establece un horizonte de desarrollo con tres objetivos específicos y acumulativos: prestar

servicios en línea, una mejora en la gestión, un empoderamiento y una mayor confianza de los ciudadanos, todo lo cual debería conducir a construir un Estado más eficiente, más transparente y más participativo”. En este orden de ideas, sería ideal que junto con los esfuerzos del gobierno sea también pertinente para las universidades implementar este tipo de políticas que permitan mejorar la entrega de sus servicios y estar a la vanguardia de la tecnología en administración.

2.4 Instituto de Proyección Regional y Educación a Distancia

El Instituto de Proyección Regional y Educación a Distancia, IPRED, creado en el Acuerdo No. 061 de 2008 del Consejo Superior, fusionando el Instituto de Educación a Distancia, INSED, y la Dirección General de Regionalización, tiene como eje misional el fortalecer la acción de la Universidad en las regiones, mediante el cumplimiento de sus funciones de formación, investigación y extensión. En su visión declara que “El Instituto contará con una eficiente y moderna infraestructura física y tecnológica, que le permitirá responder con proyección nacional e internacional a los compromisos misionales, convirtiéndose en un nodo de la actividad cultural y académica de la región.”, por lo cual se puede entender que entre sus propósitos está poseer una infraestructura tecnológica moderna que apoye sus funciones, siguiendo también el proyecto institucional de la Universidad Industrial de Santander.

El IPRED está compuesto por el Consejo de Sedes, de Instituto y de Programas, estos tres consejos toman decisiones clave en varios ámbitos académicos e institucionales. El Consejo de Programas está integrado por el Subdirector Académico y los Coordinadores de Programas impartidos por el IPRED, este consejo decide sobre asuntos académicos y asesora al subdirector

en la toma de decisiones. El Consejo de Programas se reúne semanalmente y trata temas en determinadas categorías, estos temas son evaluados y posteriormente publicados en conjunto en forma de actas, las actas son publicadas en físico para su posterior consulta. El proceso descrito anteriormente puede demorar de dos (2) semanas a un (1) mes, dependiendo del volumen de temas que comprenda cada acta, además, para su consulta, el sujeto o unidad interesada debe dirigirse a la oficina donde se guarden las actas. El aplazamiento en la publicación de actas se debe a los reprocesos en su creación y en el tratamiento de cada uno de los temas, en ocasiones hay temas que se dejan en estado de espera por varias sesiones. Adicionalmente, cada consejo posee una estructura y un proceso diferentes en la creación de actas dificultando el proceso de compartir información entre consejos. En ocasiones hay pérdida de información por la forma en cómo editan las actas y no hay manera de hacer seguimiento a las ediciones que se llevan a cabo, entre otros problemas. Es por eso que se decide crear una plataforma web donde se centralizan los datos de actas, y se pueda no solo crear y editar sino también consultar rápidamente la información correspondiente, comenzando por el Consejo de Programas del IPRED y con proyección a dar soporte a todos los consejos restantes. Esta aplicación web tendrá el nombre de Sistema de Gestión de Reuniones, SIGER.

2.5 Herramientas de desarrollo

Se planea crear una aplicación web llamada Sistema de Gestión de Reuniones, SIGER, esta podrá ser accedida desde cualquier dispositivo, será fácil de desplegar y versátil a cambios. Se decidió escoger *MongoDB* como base datos, *NodeJS* como servidor y proveedor API REST y *React* para construir la plataforma web que consuma el API y funcione como interfaz para que

tanto funcionarios del IPRED como estudiantes de carreras a distancia pueda acceder a la información correspondiente a actas de consejos de manera eficiente.

2.5.1 MongoDB. MongoDB es un sistema de bases de datos NoSQL lanzado en 2009. Guarda sus datos como documentos en formatos JSON con una estructura dinámica y es una base de datos creada con cuatro (4) objetivos clave: flexibilidad, replicación, velocidad y facilidad de uso [2]. Soporta replicación e indexado en diferentes servidores sin realizar configuraciones complicadas, haciéndola una base de datos óptima para escalabilidad. Posee controladores para múltiples lenguajes de programación. MongoDB garantiza tiempos de consulta eficientes y por ello es usada extensivamente en proyectos relacionados con minería de datos. A diferencia de las bases de datos basadas en SQL, MongoDB no soporta transacciones, operaciones de indexado complejas, ni operaciones SQL como *JOINT*, entre otros, pero estas operaciones se pueden hacer en la parte del servidor fácilmente. Se decide usar esta base de datos por su flexibilidad a los cambios, al despliegue en diferentes servidores y su facilidad de configuración.

2.5.2 Node.JS. NodeJs es un ambiente de programación en Javascript orientado a la escritura de procesos de servidor. Está basado en el motor de Javascript de Google V8, y está escrito en C y C++ por lo cual ofrece un gran rendimiento. Usa un paradigma basado en eventos en vez de multihilo para manejar procesos paralelos. Lo anterior combinado con la versatilidad y extensibilidad de Javascript permiten escribir aplicaciones orientadas a servicios de manera eficiente [14]. Node.Js cuenta con un entorno rico en librerías para diferentes propósitos y un sistema de manejo de paquetes versátil y rápido, combinado con MongoDB ofrece un entorno óptimo para la creación de servicios REST y por esto fue escogido para este proyecto.

2.5.3 React. *React* es una librería en *Javascript* creada por *Facebook* en 2013 con el propósito de resolver los retos que se presentan en el desarrollo de interfaces de usuario complejas con datos que cambian constantemente. *React* popularizó en la web el concepto de componente, el cual es un objeto especial que se renderiza en la pantalla y su apariencia cambia dependiendo de su estado y su ciclo de vida. *React* permite crear extensiones para agregar nuevas funcionalidades que disminuyen el tiempo de desarrollo de manera considerable. Este proyecto usa *React* pues permite estructurar la plataforma en componentes permitiendo un mejor mantenimiento y reutilización, además, posee un entorno de desarrollo que tiene como lineamientos las técnicas recomendadas para el desarrollo de páginas web. Es pertinente mencionar que *React* es un proyecto de código abierto y cuenta con una documentación exhaustiva que permite su rápido aprendizaje, y es usada en producción por grandes empresas como *Facebook*, *Spotify*, *Instagram*, *Whatsapp*, *Netflix*, entre otros, lo que demuestra su madurez y confiabilidad en el mercado.

2.6 REST

La transferencia de estado representacional o REST, es una arquitectura web que permite la transmisión de información definiendo una serie de reglas de comunicación, habilitando a las aplicaciones web consumir datos de servidores [4]. Esto permite separar intereses y es la base de la arquitectura cliente-servidor. En REST, existe una serie de recursos que son expuestos mediante una URI, o identificador de recursos uniforme [4]. Si se desea exponer el recurso de actas su URI de acceso según REST es “/actas”. El cliente que haga peticiones debe enviar información en la URI necesaria para que el servidor procese la petición, esta información puede incluir variables de sesión, o claves de autorización de recursos. El servidor después de procesar la petición enviará una respuesta al cliente, generalmente contendrá datos en un formato específico [4]. Cada petición enviada por el cliente debe tener una cabecera que contenga la acción a realizar junto con otras variables. Las acciones más comunes son:

- **GET:** Usada para obtener un recurso o una lista de datos.
- **POST:** Usada para crear un nuevo dato dentro de un recurso, generalmente se envía la petición con un cuerpo.
- **PUT/PATCH:** Usadas para modificar un dato dentro de un recurso.
- **DELETE:** Usado para eliminar un dato dentro de un recurso.

Este proyecto usa REST para transmitir los datos del servidor al cliente web. La aplicación web tendrá una interfaz cuyas acciones se traducirán a peticiones REST enviadas al servidor, permitiendo la administración de datos.

2.7 SCRUM

SCRUM es un conjunto de procesos de desarrollo ágil creado para adaptarse a diferentes tipos de proyectos, entregando de forma óptima productos de gran calidad. Es usado extensivamente en el desarrollo de aplicaciones software y a diferencia de otros modelos de desarrollo, realiza una serie de iteraciones en las cuales va agregando características al software, permitiendo adaptarse al cambio y a los requisitos complejos y dinámicos.

SCRUM fue construido sobre tres pilares. El primero es la transparencia, porque existe un lenguaje común y definido entre los miembros del grupo de trabajo, reduciendo los malentendidos sobre las tareas a realizar en el producto. El segundo es la inspección, porque las tareas o artefactos son frecuentemente inspeccionados para evitar errores o desvíos, reduciendo tiempo y dinero. El último es la adaptación, pues planea reuniones regulares para analizar los requerimientos y el estado de producto para determinar si la dirección a la que va es correcta y reducir errores.

El equipo SCRUM es el grupo de personas que se encargan del desarrollo del producto, consisten en:

- **El dueño del producto (*Product Owner*):** Se encarga de definir los requisitos (*Product Backlog*), optimizar el valor de trabajo, asegurarse que los requisitos sean transparentes, visibles y comprendidos por todos los miembros del equipo.
- **El equipo de desarrollo (*Development Team*):** Son el grupo de personas encargadas de ejecutar la implementación del producto. Son autoorganizados, es decir,

ellos definen el orden en que realizan las actividades También son multifuncionales, pues poseen múltiples habilidades necesarias para completar el producto.

- **Scrum Máster:** Es el líder del proyecto a nivel de SCRUM, tiene múltiples funcionalidades y sirve como puente entre otros actores y el equipo de trabajo. Promueve y es experto en la filosofía y políticas de SCRUM.

En la metodología SCRUM existen eventos que se llevan a cabo mientras se desarrolla el producto. El más importante es el *Sprint*, que constituye una unidad de tiempo, máximo de un mes, en donde se completa el objetivo, o *Sprint Goal*. El objetivo agrega una nueva característica o un conjunto de características relacionadas al producto. El *Sprint* puede ser cancelado en caso de que la dirección o nueva característica no genere el mayor valor posible, aunque este evento es poco probable ya que una cancelación desperdicia los recursos consumidos en el transcurso del *Sprint*.

Durante el *Sprint* se llevan a cabo otras tareas, la primera es la planificación del *Sprint*, en donde se define el objetivo y cómo llevarlo a cabo, para ello define una “Lista de pendientes” o *Product Backlog*. Una estrategia útil para realizar la lista de pendientes se llama Mapeo de Historias o *Story Mapping*, en donde se listan los diferentes usuarios y qué acciones pueden realizar para a partir de ellas obtener las actividades a realizar. Durante el transcurso del *Sprint* se realiza diariamente un *Daily Scrum*, en donde cada miembro comunica las actividades que realizará ese día y lo que necesita para alcanzar su objetivo. Al finalizar el *Sprint* se realiza una revisión donde el equipo muestra el trabajo terminado. El dueño del producto explica cuáles elementos del *Sprint Backlog* fueron completados y cuales falta haciendo una proyección a la siguiente iteración.

La aplicación web SIGER fue desarrollada usando esta metodología, aunque ajustada para equipos pequeños y las necesidades del IPRED. El rol de *Scrum Master* fue realizado por la Ingeniera Maria Alejandra Acevedo, el *Product Owner* fue el Profesor David Alejandro Miranda Mercado y el equipo de desarrollo fue conformado por el estudiante de Ingeniería de Sistemas Andrés Felipe Camacho Velandia.

3. Diseño de la aplicación web

El Sistema de Gestión de Reuniones, SIGER, es una aplicación web que permite la administración y consulta de la información generada en las reuniones de consejos del IPRED. En dicha reunión evalúan temas en distintas categorías, cada tema se conceptúa y se le agrega un estado, un ejemplo de estado de tema es ‘Favorable’ que indica que el tema puede ser aprobado, pero necesita alguna información adicional. Las secretarías del IPRED son las encargadas de la creación del acta y de su revisión previo a su publicación. Antes de cada reunión debe existir un borrador del acta donde se adjunten los temas que se vayan a tratar con sus respectivos archivos de soporte. Durante la reunión, los consejeros evalúan cada uno de los temas y emiten un concepto. Al finalizar la reunión, se publica un acta conformada por los temas tratados, esta tarea la realizan las secretarías. Después de publicada el acta se pueden hacer consultas sobre el estado del tema y su concepto.

En el proceso descrito anteriormente se pueden identificar tres (3) tipos de usuario que interactúan en el proceso: Secretaria, Consejero (Coordinador) y Consulta. Dados estos usuarios se puede crear el *Story Mapping*, un diagrama propio de la metodología *Scrum*, para obtener los requerimientos del producto. El *Story Mapping* es una forma concisa de mostrar los requerimientos de una aplicación y la forma de llevarlos a cabo. Aunque no obtiene todos los requerimientos del producto final, ayuda a tener una idea principal de lo que se necesita para hacer el primer prototipo funcional de esta.



Figura 1. Story Mapping de rol de secretaria de SIGER.



Figura 2. *Story Mapping* de roles de coordinador y consulta de SIGER.

Las figuras 1 y 2 muestran el *Story Mapping* para los diferentes usuarios de SIGER. En Rojo se encuentran las personas, o usuarios que pueden usar SIGER. En amarillo se encuentran las historias o posibles acciones que pueden realizar los usuarios y en verde el detalle de las historias o tareas que hay que realizar para hacer la funcionalidad.

Es necesario agregar unos comentarios adicionales al *Story Mapping* y el planteamiento de los requerimientos de SIGER. Se requiere que SIGER sea fácil de desplegar tanto en su montaje inicial en el servidor, como sus respectivas actualizaciones. Esto significa que las herramientas que se seleccionen para su desarrollo deben permitir un despliegue inmediato y fácil. SIGER debe ser una aplicación de bajo consumo de recursos y que pueda ser desplegada sin configuraciones de alta complejidad en cualquier versión de Ubuntu. En la administración de actas se debe permitir el bloqueo de su edición una vez publicada, para que no se pueda manipular la información una vez agregada al archivo de actas del IPRED, también se debe permitir esconder el contenido de algunos temas que pueden contener información sensible.

Una vez desarrollada la aplicación web SIGER, el IPRED contará con una herramienta que les entregará mayor control sobre la información generada en las reuniones, se evitarán reprocesos en la creación de actas y habrá una mayor accesibilidad de la información pues su publicación será inmediata permitiendo rápida toma de decisiones. SIGER será inicialmente usando por el Consejo de Programas del IPRED, pero se planea ser desplegado en todos los consejos una vez pase su fase beta.

Teniendo en cuenta los requisitos de SIGER se hace una tabla que describe brevemente las herramientas propuestas para el desarrollo:

Tabla 1.

Herramientas propuestas para desarrollo de SIGER.

Nombre	Nivel	Función	Justificación
MongoDB	<i>Back-end</i>	Gestor de base de datos	<ul style="list-style-type: none"> ● Fácil configuración. ● Soporte nativo para réplicas en múltiples servidores. ● Permite documento de texto de 16MB y configurable. ● Velocidad de escritura y lectura rápida. ● Permite guardar documentos con estructuras complejas. ● Soporte JSON (JavaScript Object Notation) nativo. JSON es el formato de almacenamiento y transporte de datos usando por defecto en las aplicaciones web.
Node.JS	<i>Back-end</i>	Entorno de aplicación de servidor	<ul style="list-style-type: none"> ● Posee un ecosistema de paquetes exhaustivo que facilita el desarrollo. ● Soporta JSON nativamente. ● No requiere configuraciones complejas. ● Escrito en C++, ofrece gran velocidad y consumo mínimo de recursos. ● Soporte despliegue en múltiples servidores usando PM2. ● Despliegue de cambios rápido. ● Hay compatibilidad entre versiones, esto significa que en su mayoría el código escrito para node es compatible en versiones nuevas.
PM2	<i>Back-end</i>	Gestor de procesos Node	<ul style="list-style-type: none"> ● Fácil configuración. ● Permite balanceo de carga entre múltiples instancias de procesos Node. ● Divide la carga de memoria y CPU. ● Entrega estadísticas sobre el uso de memoria y CPU y el estado del proceso. ● Muestra logs sobre el estado del proceso, útil para identificación de errores. ● Soporta múltiples sistemas operativos. ● Soporte activo a correcciones de errores. ● Código abierto.

Loopback	<i>Back-end</i>	Framework de creación de API REST para Node.	<ul style="list-style-type: none"> ● Creado por IBM. ● Código abierto. ● Soporte activo. ● Creación de API REST rápida. ● Sigue los estándares de API REST. ● Conexión a base de datos MongoDB nativa. ● Permite extensión y creación de enlaces con funcionalidades complejas. ● Estructura del proyecto intuitiva, útil para nuevos desarrolladores que piensen seguir con su desarrollo. ● Curva de aprendizaje rápido.
React	<i>Front-end</i>	Librería Javascript para realizar interfaces de usuario web.	<ul style="list-style-type: none"> ● Creado por Facebook ● Facilita la construcción de páginas ● Diseño de página web por componentes ● Permite renderización de página desde el servidor, lo que la hace más rápida en móviles ● Usado extensivamente en producción por grandes empresas ● Posee un ecosistema de desarrollo exhaustivo ● Código abierto
Admin-on- Rest/React- Admin	<i>Front-end</i>	Librería de React para administración de API REST	<ul style="list-style-type: none"> ● Crea una interfaz basada en Material Design de Google ● Permite la rápida maquetación de páginas de administración de API REST ● Permite agregar funcionalidades complejas de administración. Agregar componentes ● Cambia vista dependiendo del rol del usuario ● Soporta sesiones de usuario ● Soporte para móviles, tablets y PC ● Código abierto ● Documentación completa

Las herramientas descritas en la Tabla 1 se seleccionaron teniendo en cuenta los requerimientos establecidos y teniendo como pilares la versatilidad al cambio, velocidad y bajo consumo de recursos, y facilidad de configuración. Estas herramientas serán dispuestas en el *back-end* y *front-*

end según lo establecido en la Tabla 1. En la Figura 3 se puede observar un diseño de alto nivel de cómo se servirá y desplegará SIGER. Existirá un proceso PM2 que supervisará los procesos Node corriendo el *back-end* y el *front-end* con sus respectivas herramientas (instancia de Loopback, instancia de React). El *Back-end* tendrá un servicio de API REST que servirá como interfaz para el manejo de datos desde otras aplicaciones, este API se hará consultas a la base de datos MongoDB y luego generará una respuesta en formato JSON. Este API REST estará protegido por el protocolo JWT, o *Json Web Tokens*, un modelo de seguridad basado en *tokens* que garantiza la protección de los datos. El *Front-end* correrá en la máquina del cliente y tendrá una instancia de React con vista de administración que consultará el API REST para hacer operaciones sobre los datos. Adicionalmente existirá un servidor de Back-up o de respaldo que se encargará de hacer copias de respaldo cada determinado tiempo, esto con el propósito de disminuir el riesgo de pérdida de datos.

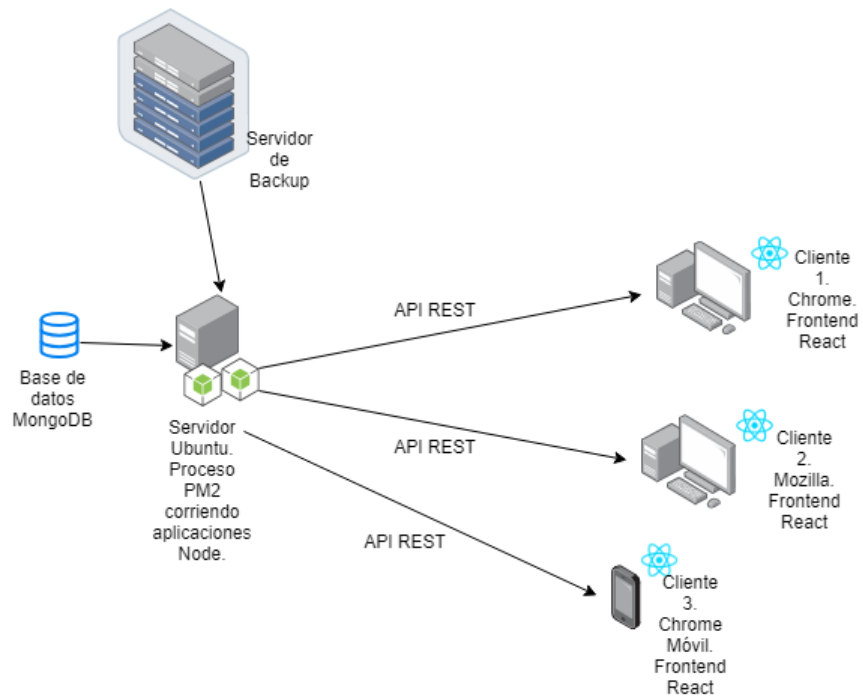


Figura 3. Diseño de alto nivel para aplicación web SIGER.

4. Estructura de datos e interfaz de usuario

En este capítulo se mostrará el diseño realizado a nivel de base de datos y páginas web.

4.1 Estructura de datos

SIGER es una aplicación web que usa API REST para la administración de sus datos, es por esto que es necesario que el formato de los datos sea JSON, pues es el lenguaje de datos estándar de la web. La estructura de estos datos consiste en un conjunto de campos clave y valor, donde el valor puede ser un número, una cadena de texto, un booleano o otro objeto JSON. Se escogió la base de datos MongoDB porque los documentos se guardan por defecto con esta estructura y pueden ser fácilmente convertidos a JSON.

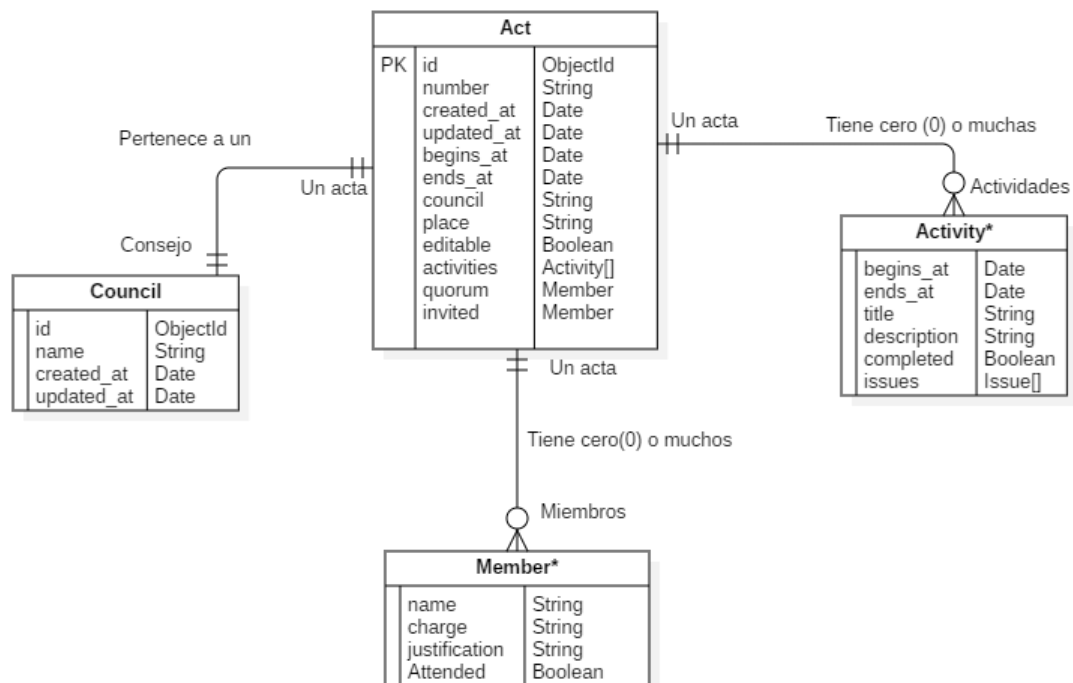


Figura 4. Modelo de datos para Acta.

Con base en los requerimientos se realizó un modelo de datos que muestra el planeamiento inicial de los datos necesarios para la creación del acta. En la Figura 4 se observa dicho modelo, las entidades con un asterisco representan modelos embebidos, esto significa que son insertados en ese campo del documento y no referenciados. El acta debe poder filtrarse por diferentes campos, es por ello que es necesario guardar información sobre número de acta (*number*), el lugar (*place*), la fecha de inicio (*begins_at*) y final (*ends_at*), y el consejo que llevó a cabo la reunión (*council*). El acta tiene un campo que indica si puede ser editada o no (*editable*), este campo será utilizado para bloquear futuras ediciones después de ser publicada. En cada reunión del Consejo de Programas se lleva a cabo una lista de actividades (*activities*) y estas son registradas en el acta, estas corresponden a las categorías en las cuales pueden ser clasificados los temas. Las actividades son:

- 1. Comprobación del quórum:** En esta actividad se lleva a cabo un repaso de la lista de participantes en la toma de decisiones del acta, entre ellos están Los Coordinadores de Programa y el Subdirector Académico. Además de la lista de invitados.
- 2. Comprobación del orden del día:** Se comprueba el orden en el cual se llevarán a cabo el resto de las actividades.
- 3. Aprobación de actas:** Se revisan las actas anteriores y se aprueban y firman en caso de ser necesario.

4. **Asuntos de la subdirección académica:** En esta categoría se agendan los temas relacionados a la subdirección académica.
5. **Asuntos académicos:** Es una categoría de tema.
6. **Asuntos de profesores:** Es una categoría de tema.
7. **Asuntos de estudiantes:** Es una categoría de tema.
8. **Asuntos de miembros del consejo:** Es una categoría de tema.

La estructura de las reuniones cambia con poca frecuencia, pero en ocasiones pueden existir casos especiales en los que se necesite agregar una actividad adicional. En la Figura 5 se muestra el modelo entidad-relación para actividad y tema, el tema corresponde la unidad más pequeña de información generada por el consejo. Este pertenece a una categoría (*type*) específica y dependiendo de esta categoría es tratado en determinado orden en la reunión de consejo. Un tema pertenece a un acta (*act*), y necesita un título (*title*), una descripción (*description*) para conocer sobre qué temática trata el tema. Esta temática requiere en ocasiones archivos de soporte (*attachments*). Al crear el tema su estado (*state*) será ‘En espera’, lo que significa que el tema será evaluado en la siguiente reunión. Una vez evaluado en la reunión el tema es conceptuado o se le agrega un concepto (*feedbacks*), el concepto es por lo general emitido por un Consejo de Programas. Los temas poseen también dos campos de control y visibilidad para determinar si son editables (*editable*) y visibles al público (*public*).

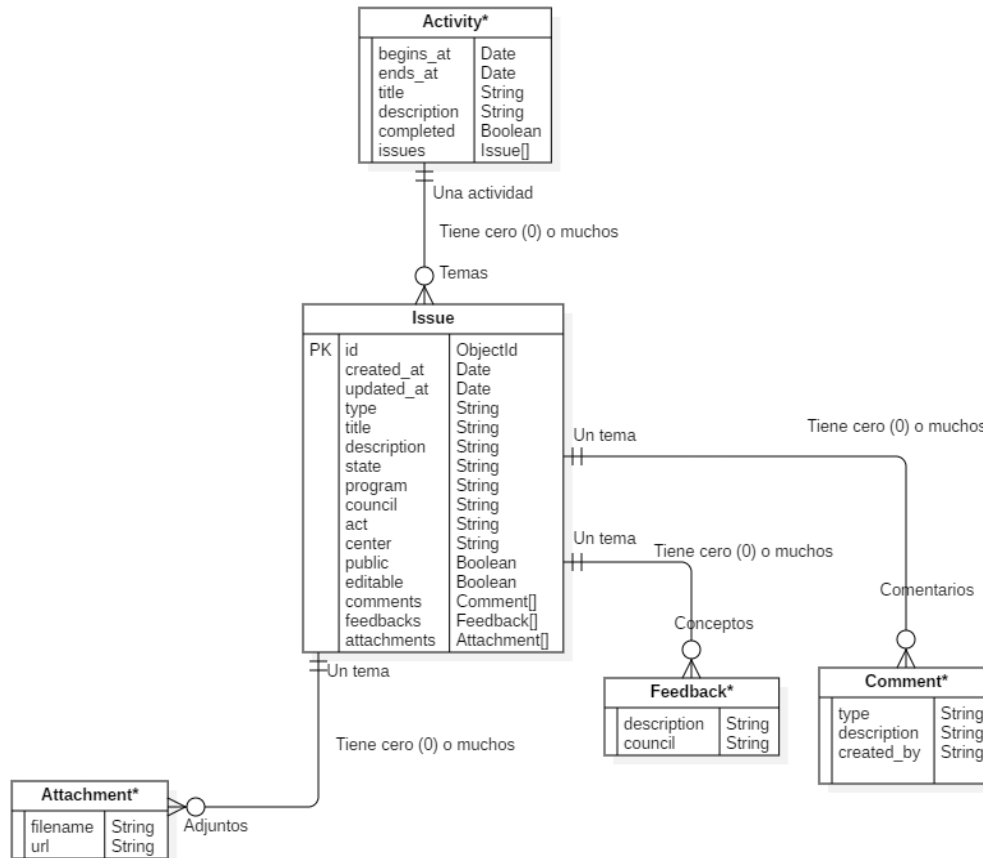


Figura 5. Modelo de datos para actividad, tema y sus respectivas relaciones.

Con este modelo de datos se podrá guardar la información necesaria para cumplir con los requerimientos, se debe tener en cuenta que el motor de base de datos MongoDB permite un esquema de modelo dinámico que puede cambiar con el tiempo de manera óptima. Hay que hacer algunas aclaraciones adicionales sobre los adjuntos (*Attachment*), el servidor también funcionará como un gestor de archivos, cuando se guarde un archivo el *back-end* genera una url única para su descarga, este url apunta a la dirección https del servidor donde esté SIGER.

4.2 Diseño de interfaz

SIGER debe permitir la administración de la información del acta y los modelos relacionados con esta, la interfaz deberá permitir crear, editar, listar y remover información del acta de manera intuitiva y rápida. La vista de edición debe ser lo suficientemente versátil para mostrar la información necesaria durante las reuniones de consejo, pues el acta se editará en tiempo real mientras se lleva a cabo la reunión. Es por eso que se escoge *Admin-on-rest* para la maquetación de la aplicación web de administración. *Admin-on-rest* es una librería para React que permite la creación de páginas de administración teniendo el API REST. A cada recurso o modelo en el API se le crea un módulo de creación, edición, listado y eliminación. En caso de que se necesite realizar un módulo personalizado la librería ofrece herramientas para realizar esto sin requerir configuraciones avanzadas. *Admin-on-rest* sigue los estándares de diseño de *Material Design*, creado por Google. *Material Design* fue creado para traer lo mejor en diseño de interfaces combinado con la posibilidad de personalizar colores y que haya un diseño coherente entre diferentes plataformas. La vista de creación de acta (Figura 6), se dividirá en cuatro (4) pestañas para condensar la información en categorías y minimizar el movimiento del cursor. Las pestañas de Quórum e Invitados servirán para agregar integrantes y marcar su asistencia. Hay algunos campos que tendrán un asterisco (*) al final, estos campos son obligatorios para poder guardar el acta. En la Figura 7 se observa la pestaña de Actividades, en las cuales se pueden agregar temas al presionar la casilla de temas. El botón listar llevará a la vista de lista de actas.

The screenshot shows the 'Nueva acta' (New Act) form in the SIGER system. The form is titled 'Nueva acta' and has a 'LISTAR' button in the top right corner. The form is divided into four tabs: 'INFORMACIÓN BÁSICA' (selected), 'QUORUM', 'INVITADOS', and 'ACTIVIDADES'. The 'INFORMACIÓN BÁSICA' tab contains the following fields:

- Número de acta *
- Fecha de inicio: Time(hours, mins.) with a clear button (X)
- Fecha de finalización: Time(hours, mins.) with a clear button (X)
- Consejo: Consejo de Programas (dropdown menu)
- Lugar: Sala de Juntas de la Dirección
- Acta escaneada: Suelte el archivo para cargar o haga click para seleccionar uno.

A 'GUARDAR' button is located at the bottom left of the form.

Figura 6. Vista de creación de Acta en SIGER.

The screenshot shows the 'Nueva acta' (New Act) form in the SIGER system, specifically the 'ACTIVIDADES' tab. The form is titled 'Nueva acta' and has a 'LISTAR' button in the top right corner. The form is divided into four tabs: 'INFORMACIÓN BÁSICA', 'QUORUM', 'INVITADOS', and 'ACTIVIDADES' (selected). The 'ACTIVIDADES' tab contains the following fields:

- Título: Asuntos Académicos
- Temas

A 'GUARDAR' button is located at the bottom left of the form.

Figura 7. Vista de creación de acta, pestaña de actividades en SIGER.

Actas

≡ BUSCAR + CREAR ↻ ACTUALIZAR

Año
2019-01-01

NÚMERO DE ACTA	FECHA DE INICIO	FECHA DE FINALIZACIÓN	PDF
14	5/8/2019, 7:00:00 AM	5/8/2019, 9:00:00 AM	EDITAR
13	4/23/2019, 8:00:00 AM	4/23/2019, 10:00:00 AM	EDITAR
12	4/12/2019, 10:00:00 AM	4/11/2019, 12:00:00 PM	EDITAR
11	4/2/2019, 8:00:00 AM	4/2/2019, 10:00:00 AM	EDITAR
10	3/28/2019, 10:00:00 AM	3/28/2019, 12:00:00 AM	EDITAR
09	3/21/2019, 10:00:00 AM	3/21/2019, 12:00:00 PM	EDITAR
08	3/12/2019, 8:00:00 AM	3/12/2019, 10:00:00 AM	EDITAR
07	3/5/2019, 8:00:00 AM	3/5/2019, 10:00:00 AM	Acta N° 7.pdf EDITAR
06	2/26/2019, 8:00:00 AM	2/26/2019, 10:00:00 AM	EDITAR
05	2/19/2019, 8:00:00 AM	2/19/2019, 10:00:00 AM	EDITAR

1-10 de 14

1 2 SIGUIENTE >

Figura 8. Vista de lista de actas creadas en SIGER.

La vista de lista (Figura 8) ofrecerá un resumen de las actas que han sido creadas, y las fechas en las cuales se realizaron, además poseerá un campo para ver el acta firmada en PDF. Las actas se podrán organizar de mayor o menor o viceversa presionando el texto “NÚMERO DE ACTA”, esta misma funcionalidad está para todos los textos en la cabecera de la lista. El botón editar llevará a la vista de edición.

Figura 9. Vista de edición de Acta.

La vista de edición de acta (Figura 9) es similar a la vista de creación, pero posee la pestaña de “ORDEN DEL DÍA”, es esta pestaña aparecerán las actividades que se llevarán a cabo y se podrá cambiar su orden de acuerdo con la decisión del Consejo de Programas. Adicionalmente aparece las opciones de exportar y eliminar el acta. Las vistas de creación, lista y edición son similares para el resto de los recursos en el API. Adicionalmente, existirá un menú donde se pueden seleccionar los recursos para administrar cada uno de ellos (Figura 10). Al seleccionar cualquier opción se verá la lista de ese recurso y desde ahí se podrá crear y editar.

Sistema de gestión de reuniones - Consejo de Programas

Actas
Temas
Programas
Consejos
Sedes
Tipos de comentarios

Editar acta

EXPORTAR LISTAR ELIMINAR ACTUALIZAR

INFORMACIÓN BÁSICA QUORUM INVITADOS ORDEN DEL DÍA ACTIVIDADES

Número de acta *
14

Fecha de inicio
2019-05-08 7 am

Fecha de finalización
2019-05-08 9 am

¿Editable?

Consejo
Consejo de Programas

Lugar
Sala de Juntas de la Dirección

Acta escaneada
Suelte el archivo para cargar o haga click para seleccionar uno.

GUARDAR

Figura 10. Vista de SIGER. A la izquierda menú de recursos. A la derecha vista de edición de acta.

5. Descripción de los módulos diseñados

SIGER es una aplicación diseñada por módulos debido a esta arquitectura permite cambiar y mantener el proyecto de forma organizada y se pueden introducir cambios sin alterar la estructura del proyecto, cada módulo tendrá una tarea específica y puede ser removido o reemplazado en caso de ser necesario sin afectar el proyecto completo. Los módulos de SIGER son:

5.1 Módulo de Back-End

El *back-end* hace la función de puente entre la base de datos MongoDB y el cliente mediante una interfaz de aplicación con lenguaje REST. Un API REST consiste en una serie de peticiones HTTP que realizan acciones sobre los datos, estas acciones son crear, modificar, eliminar, consultar, entre otras. Este módulo está compuesto por submódulos con tareas específicas, que se conectan al controlador del API REST (Figura 11), estos son el módulo de autenticación, almacenamiento y módulo de datos.

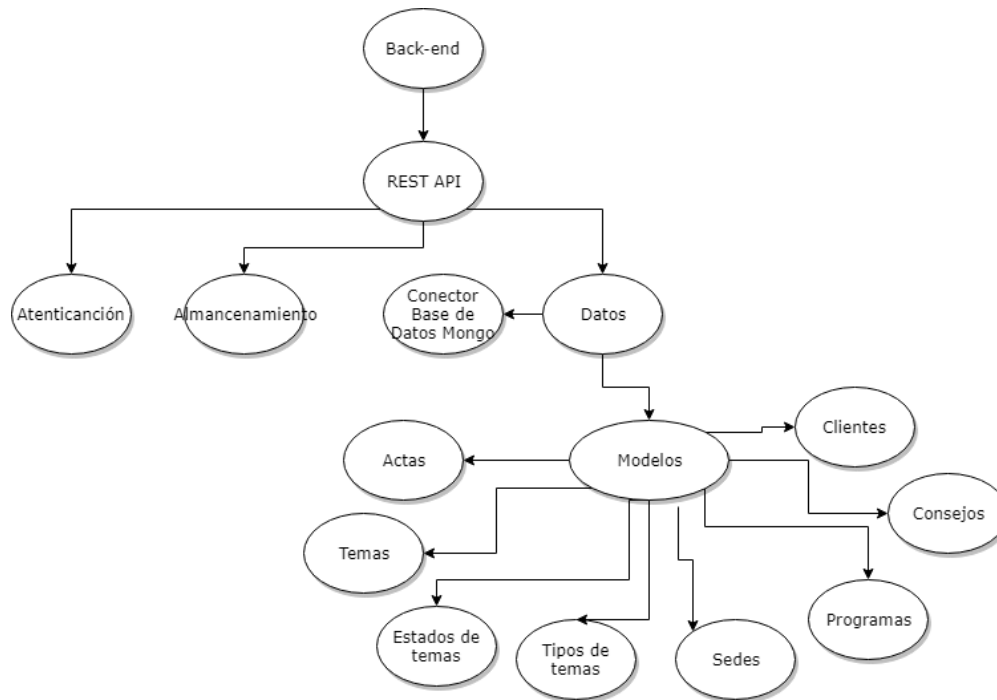


Figura 11. Diagrama de módulos del Back-end.

5.1.1 Módulo de autenticación. La principal tarea del módulo de autenticación es prevenir el acceso a recursos no autorizados mediante el uso de la estrategia Json Web Tokens, en la cual antes de hacer cualquier petición el usuario deberá pedir un token con único generado a partir de sus credenciales, en el caso de SIGER estas son el correo y la contraseña. Una vez se tiene el token, cada petición que se haga deberá contener en la cabecera esta información y así en el back-end se verifica si el token tiene acceso al recurso que está siendo requerido y de no ser así rechazar la petición.

5.1.2 Módulo de Almacenamiento. La tarea del módulo de almacenamiento es guardar los archivos de soporte de temas en varios formatos y las actas firmadas en formato PDF. Este módulo permite guardar archivos en diferentes proveedores en la nube o en algún directorio ubicado en el servidor, por defecto, se guardan los archivos en “/storage/container” dentro del directorio del back-end.

5.1.3 Módulo de Datos. Este módulo se encarga de definir los modelos que serán guardados en la base de datos y generar un controlador para administrarlos mediante la REST API. Para definir un modelo de datos se necesitan dos archivos, uno en formato JSON (Figura 12) donde se definen los campos que tiene el modelo, las validaciones para los campos, el tipo de campo, las variables de control de acceso en caso de que se restrinja para algunos roles de usuario y sus relaciones con otros modelos. Las relaciones con otros modelos pueden ser de tipo “incluye a”, “referencia uno”, “referencia muchos”, entre otros.

Adicionalmente existe otro archivo de modelo de datos en lenguaje Javascript, en este archivo se permite modificar las acciones del controlador REST para agregar acciones más complejas, como una búsqueda indexada o el cálculo de alguna variable a partir de los campos del modelo.

En este módulo también se define el controlador de la base de datos MongoDB, este archivo está en formato JSON y en él se definen las variables necesarias para hacer una conexión a la base de datos Mongo, puede cambiar dependiendo del entorno en que se ejecute ya sea producción o pruebas.

```
{
  "name": "Act",
  "plural": "acts",
  "base": "PersistedModel",
  "strict": "filter",
  "idInjection": true,
  "options": {
    "validateUpsert": true
  },
  "mixins": {
    "Timestamp": true
  },
  "properties": {
    "number": {
      "type": "string"
    },
    "begins_at": {
      "type": "date"
    },
    "ends_at": {
      "type": "date"
    },
    "council": {
      "type": "string"
    },
    "place": {
      "type": "string"
    },
    "editable": {
      "type": "boolean",
      "default": true
    }
  }
}
```

Figura 12. Definición de modelo de datos de acta con archivo JSON.

Los modelos guardados en el módulo de datos son:

- **Acts:** Modelo de acta. Los datos sólo pueden ser accedidos por un usuario autenticado. Este modelo define una acción de búsqueda avanza usado en la vista de consulta, donde se busca por palabra clave y devuelve una lista de selecciones donde esté incluida esa palabra y su posición dentro del tema. También hay una acción de exportar el acta en PDF directamente.
- **Issues:** Modelo de temas. Tiene indexados los campos de título, descripción y conceptos para su búsqueda basada en palabra clave.
- **IssueStates:** Modelo de estados de tema, identifica la fase de evaluación en la que se encuentra el tema, estos son:
 - a.** No Favorable
 - b.** Favorable
 - c.** En Espera
 - d.** Aprobado
 - e.** No Aprobado
 - f.** Aplazado
 - g.** Tratado
 - h.** Enterado
 - i.** Inhibitorio
- **IssueTypes:** Modelo de tipos de tema, estos determinan la categoría a la cual pertenece el tema y la prioridad que tiene a la hora de su evaluación en la reunión de consejo, estos tipos son:

- a. Estudiantes
 - b. Subdirección académica
 - c. Académicos
 - d. Profesores
 - e. Miembros del consejo
 - f. Varios
- **Programs:** Modelo de programas. Este modelo es utilizado en el tema y determina de qué programa académico proviene, generalmente aparece cuando es un tema de tipo “Estudiantes”, los programas son:
 - a. Producción Agroindustrial
 - b. Tecnología Agroindustrial
 - c. Técnico Profesional en Producción Agropecuaria
 - d. Gestión Empresarial
 - e. Agro
 - f. Introductorio
 - g. Ciclo básico
 - h. Ingeniería Forestal
 - i. Zootecnia
 - j. Turismo
- **Clients:** Este modelo es usado para guardar la información de los usuarios de SIGER y sus respectivos roles, dependiendo del rol cambia la vista de la aplicación y se restringen algunas acciones. Los roles de usuario son:

- a. Administrador:** Tiene acceso a realizar cualquier acción en todos los modelos de datos. Este usuario crea otros usuarios y les asigna roles.
- b. Secretaria:** Administra actas y temas sin están permitido. También se encarga de alimentar la información de programas, consejos, sedes, tipos de temas y estados de temas, entre otros.
- c. Coordinador:** Posee el mismo acceso que el rol de Secretaria, pero adicionalmente puede conceptuar los temas.
- d. Consulta:** Tiene acceso de lectura en el modelo de actas, y puede hacer búsquedas por palabra clave y número de tema. Tiene acceso restringido a los modelos restantes.

5.2 Módulo de Front-End

El *front-end* constituye la parte visual de SIGER, su arquitectura, al igual que el back-end, está basada en módulos con tareas definidas. Está escrito en Javascript utilizando React, un framework de creación de interfaces de usuario. Las interfaces de administración de recursos están hechas usando la librería Admin-on-Rest y usa el módulo de back-end para su funcionamiento, pero necesita la ayuda de algunos módulos para interconectar estos servicios. Los módulos del front-end son (Figura 13): Módulo de cliente REST, Módulo de cliente de Autenticación, Módulo de almacenamiento, Módulo de ingreso y consulta, Módulo de Interfaz de Usuario, Módulo de Exportación, Módulo de Lenguajes y finalmente Módulo de Recursos.

Previo a la descripción de los módulos es importante entender cómo funciona Admin-on-Rest. Al igual que la mayoría de librería en React, Admin-on-Rest proporciona un componente al cual

se le agregar parámetros de configuración en forma de etiquetas, similar a el lenguaje HTML. Un ejemplo de parámetro es el lenguaje (*lang*), el cual determina el idioma en que se mostrarán los componentes de la aplicación. El componente Admin-on-Rest se le agregan componentes hijos determinados recursos, cada recurso tiene parámetros para habilitar las vistas de administración, como creación o eliminación de algún dato. Cada módulo del front-end corresponde a un parámetro que se pasa al componente Admin-on-Rest para configurarlo a las necesidades de SIGER. Para su funcionamiento, Admin-on-Rest genera una serie de acciones que son recibidas por los componentes hijos y estos generan respuestas que pasan a ser el estado global de toda la aplicación, este estado global contiene los datos necesarios para alimentar las vistas de administración. Un ejemplo de esta interacción acción-respuesta se da para verificar si hay un usuario activo, para ello Admin-on-Rest genera una acción de “Verificar Sesión”, esta acción la recibe el Módulo de Autenticación el cual devuelve los datos relacionados con la sesión de usuario, si hay algún dato se muestra la vista principal, de lo contrario se muestra la vista de ingreso.

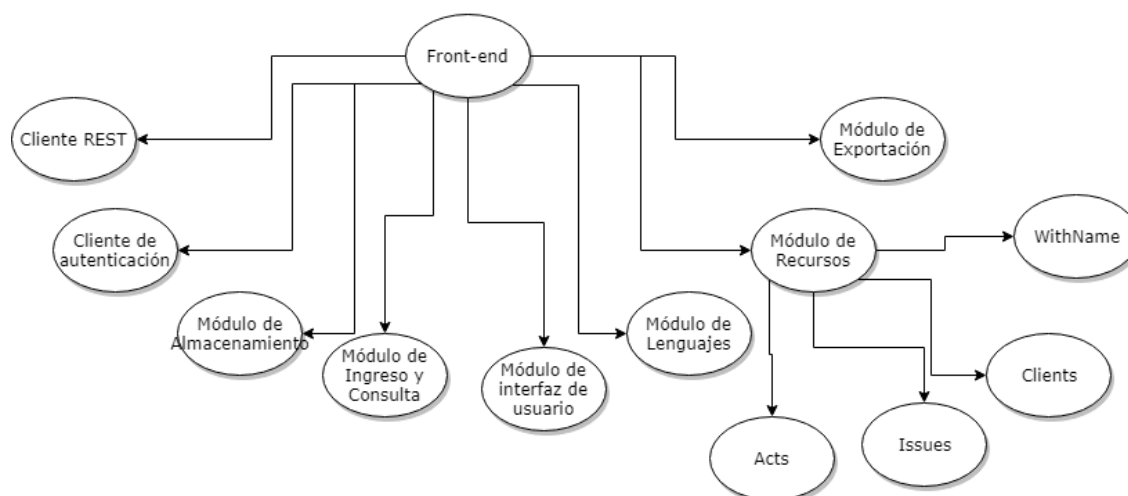


Figura 13. Diagrama de módulos del front-end.

5.2.1 Módulo de Cliente REST. El cliente REST se encarga de convertir acciones emitidas por la librería Admin-on-REST relacionada con la obtención de datos en peticiones REST al back-end. También se encarga de transformar las respuestas del back-end en un formato que pueda entender la librería para tomar decisiones y alimentar las vistas de administración. En este módulo se decide el formato del API REST y las respuestas que este emite, también agrega cabeceras como el token de autenticación para los recursos protegidos.

5.2.2 Módulo de Cliente de Autenticación. El cliente de autenticación se encarga de la lógica de sesión de usuario. Admin-on-Rest emite acciones para comprobar si hay una sesión activa o para obtener el rol del usuario en la sesión o su token de autenticación, entre otra información. Este módulo se comunica con el módulo de almacenamiento pues ahí se guarda toda esta información persistente.

5.2.3 Módulo de Almacenamiento. Este módulo se encarga de guardar datos persistentes en la sesión. En Admin-on-Rest existen dos tipos de almacenamiento, el no persistente o estado global, que contiene la información obtenida de las peticiones al back-end, información de formularios de creación y edición, entre otros. El segundo tipo es el persistente que incluye la información relacionada con la sesión de usuario, su identificador, su token de autenticación y su información básica de usuario, entre otros.

5.2.4 Módulo de Ingreso y Consulta. Este módulo es la primera vista que se observa cuando se accede a la aplicación web SIGER, contiene la lógica de inicio de sesión y consulta. La vista de consulta (Figura 14) es la primera en mostrarse, muestra la lista de actas creadas, permite la búsqueda de actas por palabras clave y número de acta. Cuando se busca por palabra clave muestra una selección de las partes del acta donde aparezca la palabra permitiendo una búsqueda rápida. La vista de consulta ofrece un botón para ingresar, al presionarlo redirecciona a la vista de ingreso (Figura 15).

Sistema de gestión de reuniones INGRESAR

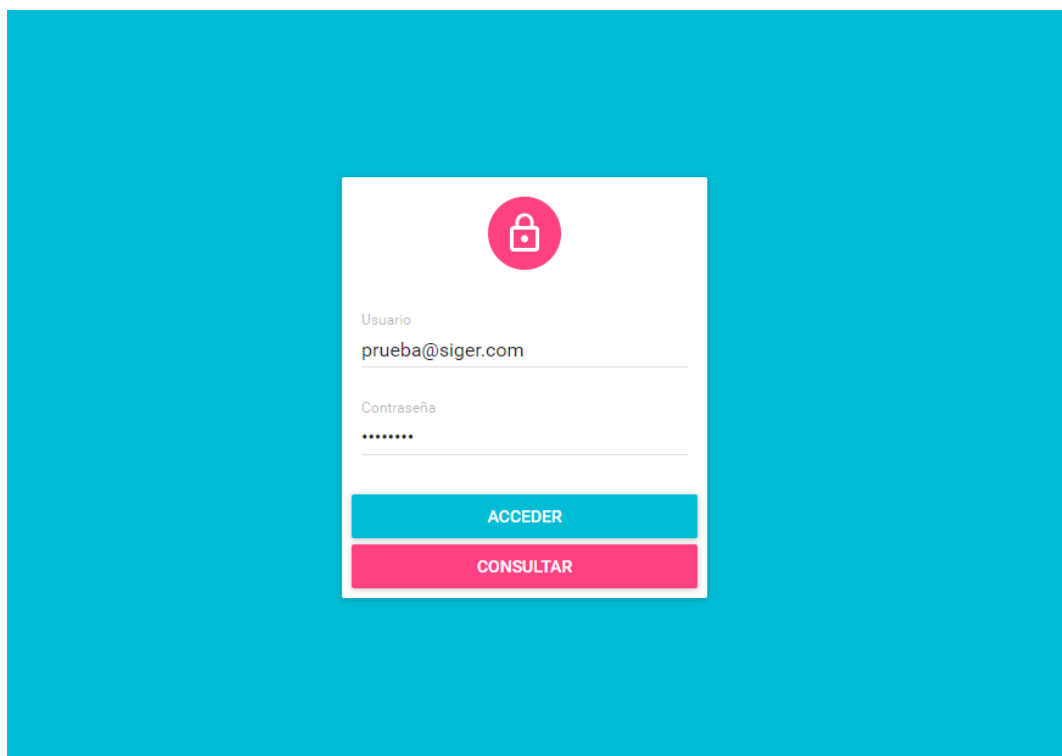
Actas BUSCAR ACTUALIZAR

Año
2019-01-01

NÚMERO DE ACTA	FECHA DE INICIO	FECHA DE FINALIZACIÓN	PDF	
14	5/8/2019, 7:00:00 AM	5/8/2019, 9:00:00 AM		MOSTRAR
13	4/26/2019, 8:00:00 AM	4/23/2019, 10:00:00 AM		MOSTRAR
12	4/12/2019, 10:00:00 AM	4/11/2019, 12:00:00 PM		MOSTRAR
11	4/2/2019, 8:00:00 AM	4/2/2019, 10:00:00 AM		MOSTRAR
10	3/28/2019, 10:00:00 AM	3/28/2019, 12:00:00 AM		MOSTRAR
09	3/21/2019, 10:00:00 AM	3/21/2019, 12:00:00 PM		MOSTRAR
08	3/12/2019, 8:00:00 AM	3/12/2019, 10:00:00 AM		MOSTRAR
07	3/5/2019, 8:00:00 AM	3/5/2019, 10:00:00 AM	Acta N° 7.pdf	MOSTRAR
06	2/26/2019, 8:00:00 AM	2/26/2019, 10:00:00 AM		MOSTRAR
05	2/19/2019, 8:00:00 AM	2/19/2019, 10:00:00 AM		MOSTRAR

1-10 de 14 1 2 SIGUIENTE >

Figura 14. Vista de consulta en SIGER.



Logo de un candado en un círculo rojo.

Usuario
prueba@siger.com

Contraseña

ACCEDER

CONSULTAR

Figura 15. Vista de ingreso en SIGER.

5.2.5 Módulo de Interfaz de Usuario. Este módulo define todos los componentes visuales, escritos en React, que se usan en la aplicación web. Desde los campos de texto hasta las vistas completas de administración de recursos con sus estados internos.

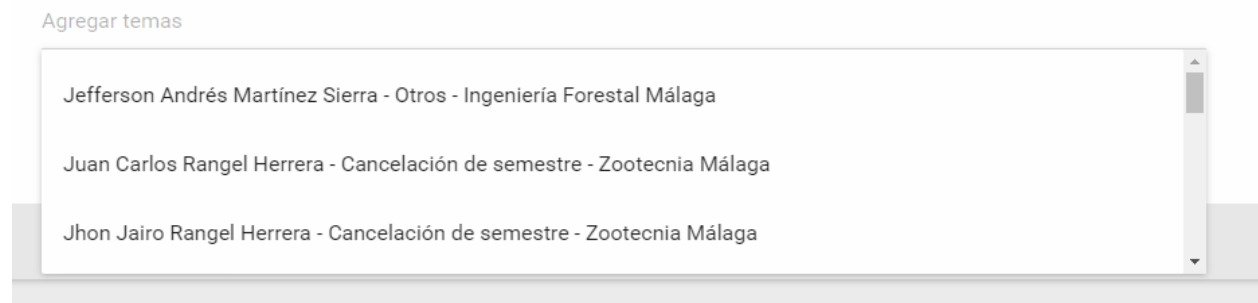


Figura 16. Componente de agregar temas.

5.2.6 Módulo de Recursos. Este módulo usa los componentes definidos en el módulo de interfaz de usuario para crear las vistas de administración de cada uno de los recursos. Estos son añadidos como componentes hijos en el componente principal de Admin-on-Rest como lo explicado en la introducción de esta sección. Los 4 recursos principales son:

- **Acts:** Vista de administración de actas
- **Issues:** Vista de administración de temas
- **Users:** Vista de administración de usuarios y roles, solo disponible para el usuario administrador
- **WithName:** Vista genérica para recursos que tengan un solo campo de nombre, como ocurre con Programas, Sedes, Tipos de Tema, Estados de Tema entre otros.

5.2.7 Módulo de exportar. Este módulo define un componente para exportar el acta o tome en formato Microsoft Word. Para realizar este proceso, primero hace una petición REST al back-end y usa la información para alimentar una plantilla con el formato de los temas del IPRED. Este módulo usa una librería de Javascript llamada *docx-templates*, cuya función es generar documentos combinando datos en formato JSON y una plantilla en formato Microsoft Word.

5.2.8 Módulo de Lenguajes. Este módulo contiene las diferentes traducciones de SIGER en múltiples idiomas, consisten en archivos en formato JSON que define parámetros de traducción. SIGER soporta inglés y español, siendo este último el idioma por defecto.

6. Algoritmos implementados

SIGER se apoya de una serie de algoritmos para su implementación, aunque algunos son propios de las librerías que se usan para su desarrollo, como el algoritmo de reconciliación usado en React para determinar qué partes de un componente se actualiza cuando cambia el estado, estos algoritmos no son creados dentro de SIGER por lo cual no serán mencionados. En esta sección se describirán cuatro (4) algoritmos implementados para SIGER, estos son el Algoritmo de Autenticación y Obtención de Token, el Algoritmo de Petición REST, el Algoritmo de Exportar Acta y finalmente el Algoritmo de Búsqueda de Temas por Palabra Clave.

6.1 Algoritmo de Autenticación y Obtención de Token.

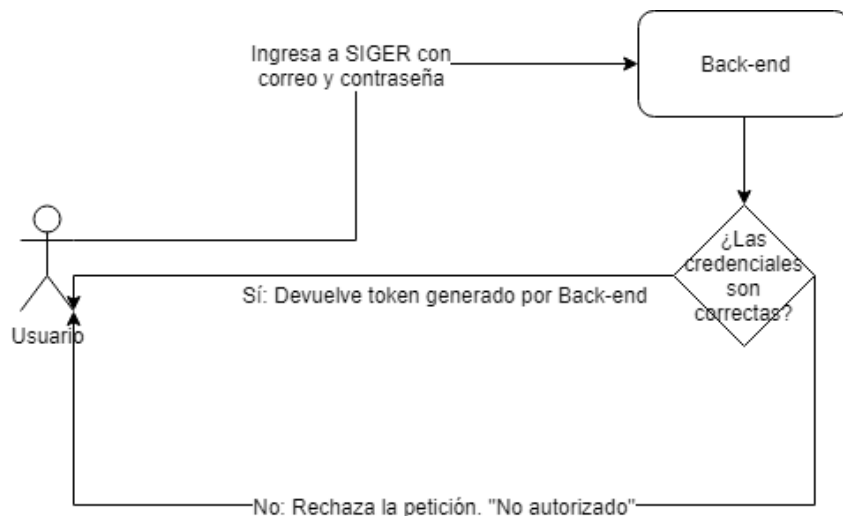


Figura 17. Diagrama de algoritmo de autenticación y obtención de token.

Este procedimiento determina la lógica de autenticación en el back-end de SIGER y la obtención de un token o una palabra de seguridad para acceder a recursos protegidos. La construcción del token está definida en el estándar RFC 7519, este documento dice que el token es el resultado de concatenar tres cadenas de texto que contiene información encriptada necesaria para procesarlo en el servidor y determinar su validez. El token contiene información del algoritmo que se usa para encriptar, información de la sesión de usuario y una firma única. El token habilita una forma de navegación en donde la información de sesión de usuario no se guarda en el servidor, común en PHP, sino en el cliente.

Cuando el cliente quiere iniciar sesión en SIGER, envía su correo y contraseña al *back-end* a través de una petición REST, si las credenciales son correctas, se pasa esta información junto con el identificador (id) de usuario para construir el token el cual se envía al cliente para que lo use al hacer peticiones. Si el correo y la contraseña no son válidos se rechaza la petición con un mensaje de “No autorizado”.

6.2 Algoritmo de petición REST

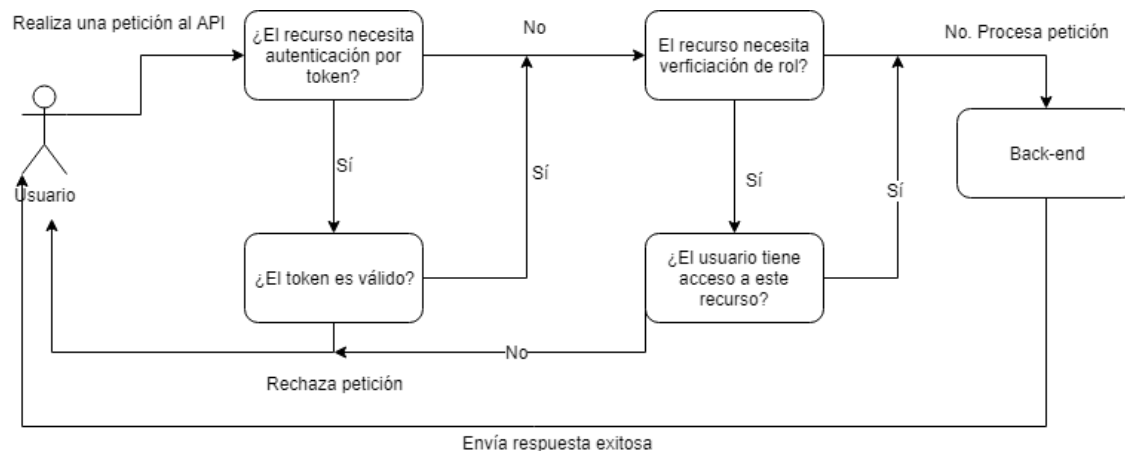


Figura 18. Diagrama de algoritmo de petición REST.

Una vez el usuario obtiene el token puede hacer peticiones web a los recursos del *back-end* de SIGER, el formato que posee estas peticiones es conocido como REST, el cual fue creado para permitir la comunicación entre clientes y aplicaciones web.

El algoritmo determina la seguridad adicional que hay al hacer cada petición, existen recursos del back-end que están protegidos por una o dos capas de seguridad, o ninguna. Por ejemplo: listar los programas no representa ningún riesgo de seguridad, por lo tanto, no tiene validaciones de token ni de rol, por otro lado, la creación de programas solo puede ser hecha por un usuario autenticado y que tenga rol de administrador, secretaria o coordinador.

6.3 Algoritmo de Exportar Acta

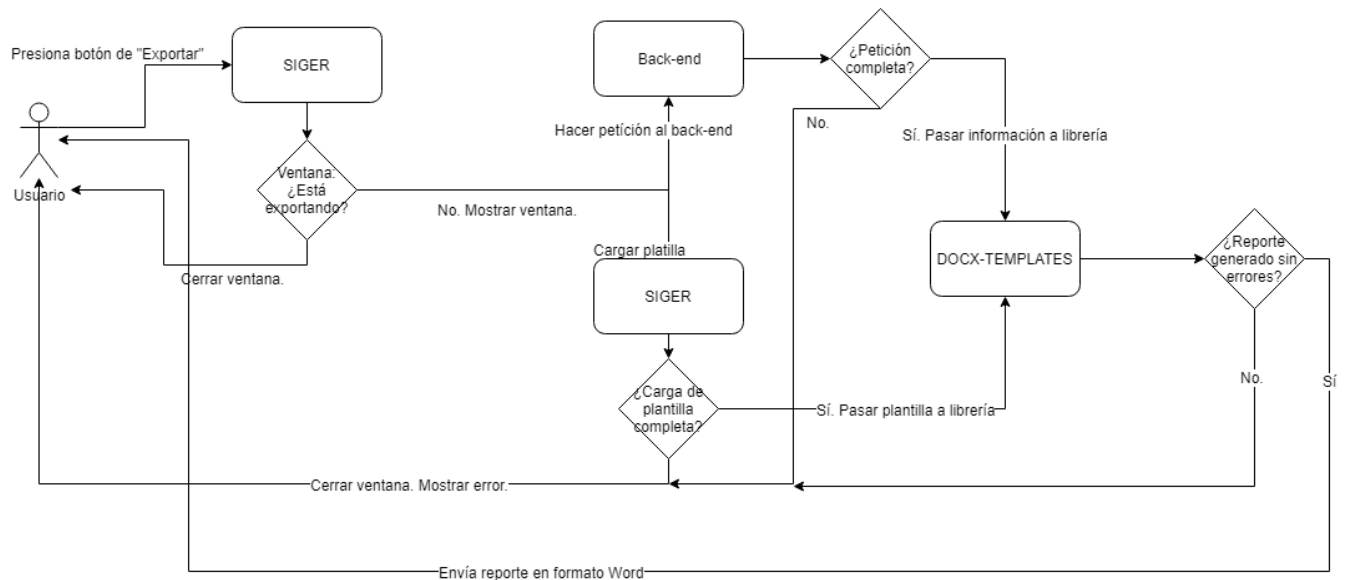


Figura 19. Diagrama de algoritmo de exportar acta.

SIGER fue desarrollado para reducir los reprocesos presentados en la creación de actas de reuniones de consejo, por lo cual es importante describir el algoritmo realizado para su exportación. La operación de exportación es realizada en el cliente y no en el servidor. El acta resultante está en Microsoft Word ya que ofrece mayor flexibilidad para las secretarías cuando otras unidades del IPRED requieren copias inmediatas del acta completa o con algunos temas en específico.

El algoritmo inicia cuando el usuario presiona el botón de exportar, este se encuentra en la vista de edición de acta y la vista de edición de tema. Al presionarlo, se muestra una ventana que indica que se está ejecutando la acción de exportar, previniendo que el cliente ejecute dos tareas de exportación al mismo tiempo. Después de mostrar la ventana, SIGER hace dos tareas paralelas, por un lado hace una petición al *back-end* para traer la información del acta, junto con los temas

seleccionados, esto depende de donde sea presionado el botón de exportar. Desde la vista de edición de acta trae el acta con todos los temas, y desde la vista de edición de tema trae el acta junto con la información de un solo tema. La segunda tarea que ejecuta paralelamente es cargar la plantilla de acta. Después de completar las dos tareas, se envía los datos de acta en formato JSON y el archivo binario a la librería *docx-templates*, la cual genera el acta final que es enviada al cliente. En caso de que falle la carga del archivo, la petición REST para traer la información del acta o el ensamblaje del acta a partir de la plantilla, SIGER cierra la ventana y muestra el error que ocurrió.

6.4 Algoritmo de búsqueda de temas por palabra clave

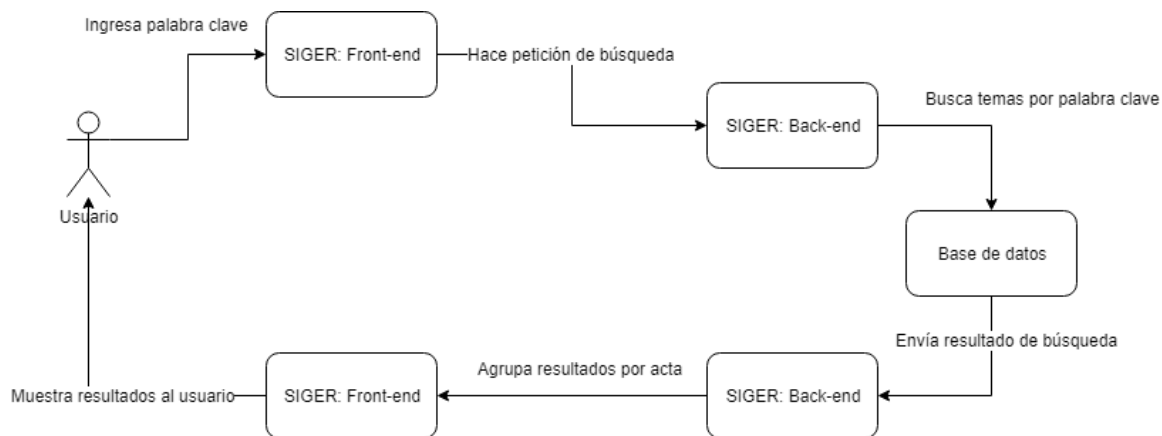


Figura 20. Diagrama de algoritmo de búsqueda de temas por palabra clave.

Uno de los requerimientos de SIGER determina que la aplicación debe permitir búsquedas por palabra clave en actas, esto significa que si el usuario necesita saber el estado de su tema, pueda ingresar su nombre y aparezcan las actas y temas que lo contienen. El desarrollo de esta característica representa un reto porque una de las limitaciones de MongoDB es que no permite búsquedas de texto de documentos relacionados por un identificador. El modelo de acta se planteó

para permitir la creación de temas en un módulo separado y luego ser agregados dentro del acta referenciados por un identificador(id). En una base de datos SQL consistiría en hacer un JOINT entre las tablas de actas y temas, pero Mongo no soporta esta operación, alternativamente ofrece un sistema de agregaciones, pero es ineficiente cuando hay una gran cantidad de temas. El algoritmo de búsqueda aprovecha la rapidez de MongoDB para búsquedas de texto simple, y pasa a hacer las búsquedas de adentro hacia afuera, esto es, de temas a actas. Debido a que la mayoría de información clave para la búsqueda se encuentra en el tema, ya que el título, la descripción y los conceptos describen mayoritariamente el tema, entonces se indexan estos campos para búsqueda de texto y se ejecuta la búsqueda. Luego de obtener el resultado se agrupa por actas, para ellos cada vez que se agregue un tema al acta se debe agregar también el identificador del acta en el tema, una vez agrupado se envía esta información al *front-end* de SIGER, este formatea los datos y los muestra al cliente. La vista de consulta usa este algoritmo, al obtener los resultados subraya en amarillo donde aparezca la palabra clave (Figura 21).

Sistema de gestión de reuniones INGRESAR

Actas BUSCAR ACTUALIZAR

Año: 2019-01-01 Palabra clave: maria

NÚMERO DE ACTA	FECHA DE INICIO	FECHA DE FINALIZACIÓN	PDF
12	4/12/2019, 10:00:00 AM	4/11/2019, 12:00:00 PM	MOSTRAR
Actividad: Asuntos de Estudiantes Tema: READMISIÓN PARA EL 2019 - 2 MARIA CATALINA GONZALEZ S...			
Actividad: Asuntos de Estudiantes Tema: Simultaneidad Maria Fernanda Pineda Mora - Maria Fernanda Pineda Mor...			
Actividad: Asuntos Académicos Tema: Situación Grados • •			
11	4/2/2019, 8:00:00 AM	4/2/2019, 10:00:00 AM	MOSTRAR
Actividad: Asuntos de Profesores Tema: Solicitud autorización profesor cátedra Escuela de Ingeniería de Petróleos • • Solicitud de eprobac... Maria identificada con L...			
09	3/21/2019, 10:00:00 AM	3/21/2019, 12:00:00 PM	MOSTRAR
Actividad: Asuntos de Estudiantes Tema: AMPLIACIÓN DEL PLAZO PARA LA LEGALIZACIÓN DE LA MATRICULA PARA EL 2019-1			

Figura 21. Resultado de algoritmo de búsqueda por palabra clave usando “Maria”.

7. Evaluación y pruebas de la aplicación web

En este capítulo se usa un modelo de evaluación basada en criterios, en este tipo de modelos se responden unas preguntas para determinar si la aplicación web cumple con unos criterios definidos. El modelo está basado en la guía de evaluación creado por el Instituto de Sostenibilidad del Software [6], que a su misma vez se basa en el *ISO/IEC 9126-1* de Ingeniería del Software para determinar la calidad de un producto, pero ajustado para aplicaciones web, ya que estas solo corren a través de un navegador y no es entregada en archivos binarios y considerando que SIGER fue creado con el propósito de ser usando en el IPRED y por un grupo de desarrollo pequeño.

Esta evaluación consiste en una serie de preguntas que ayudan a auditar el estado del software, esto con el propósito de mejorar las falencias de muestra. La evaluación por criterios está agrupada de la siguiente manera.

Tabla 2.

Tabla de criterios de evaluación agrupada.

Criterio	Sub-criterio	Notas
Usabilidad	Comprensibilidad	¿Se entiende fácilmente?
	Documentación	¿La documentación es exhaustiva, apropiada y bien estructurada?
	Capacidad de compilación	¿Es fácil compilar en un sistema operativo soportado?
	Capacidad de instalación	¿Es fácil instalar un sistema operativo soportado?
	Curva de aprendizaje	¿Es fácil de aprender a usar?
Sostenibilidad y mantenibilidad.	Derechos de autor	¿Es fácil ver quien posee el proyecto/software?
	Portabilidad	¿Puede ser usado en múltiples plataformas?
	Capacidad de soporte	¿Hay evidencia de actual o futuro soporte por parte del desarrollador?
	Analizabilidad	¿Es fácil de entender el código fuente?

Ahora se presentará un detalle de la tabla anteriormente descrita con preguntas en cada categoría para determinar la evaluación del software.

7.1 Usabilidad

Esta categoría determina qué tan fácil es entender el producto de software, su objetivo de usuarios, las funciones básicas y avanzadas. Además de la robustez de la documentación, la facilidad de compilación, despliegue y aprendizaje.

7.1.1 Comprensibilidad. La comprensibilidad responde qué tan fácil es de comprender la aplicación web para nuevos usuarios. En el caso de SIGER se muestra una alta comprensibilidad demostrando que es una aplicación que tiene un propósito claro y definido.

Tabla 3.

Tabla de criterios de comprensibilidad.

Comprensibilidad	Sí/No, Justificación si es necesario
Descripción del software de alto nivel de para qué y para quién está disponible.	Sí.
Descripción de alto nivel de lo que hace el software está disponible.	Sí.
Descripción de alto nivel de cómo funciona el software está disponible.	Sí.
La descripción de la arquitectura, con diagramas, está disponible.	Sí.
Las descripciones de los casos de uso previstos están disponibles.	Sí. Los casos de uso están descritos en el <i>Story Mapping</i> de SIGER.
Los casos de uso están disponibles.	No. La descripción proporcionada por el <i>Story Mapping</i> es concisa y suficiente para el alcance del proyecto.

7.1.2 Documentación. Este ítem determina la calidad de la documentación de la aplicación web, SIGER posee una documentación no tan exhaustiva, pero útil para sus usuarios pues muestra sus funciones básicas necesarias para empezar a usarlo. En el anexo C se encuentra el manual de usuario de SIGER que comprende las tareas básicas de creación, edición y consulta de actas y otros recursos.

Tabla 4.

Tabla de criterios de documentación.

Documentación	Sí/No, Justificación si es necesario
Proporciona una visión general de alto nivel del software.	Sí.
Particionado en secciones para usuarios, usuarios desarrolladores y desarrolladores (según el software).	Sí. La documentación está escrita para todo tipo de usuario que usa la aplicación web, debido a que la interfaz es intuitiva y ofrece las mismas características a los usuarios. Lo único que cambia entre un rol y otro es el acceso a algunos recursos. La documentación existente para desarrolladores es la correspondiente a los requisitos de instalación, herramientas requeridas, y manual de instalación.
Requiere conocimiento de fondo o experticia en algún campo	No. La aplicación web está dirigida a secretarías y coordinadores del IPRED, pero no requieren ningún conocimiento avanzado pues es solo crear y administrar datos de actas.
Enumera recursos para más información.	No. La documentación es autocontenida y no requiere conocimiento en alguna otra herramienta. Además, trata de ser concisa en cada tema desarrollado.
Es orientado a tareas.	Sí.
Consiste en instrucciones claras, paso a paso.	Sí.
Da ejemplos de lo que el usuario puede ver en cada paso, p. Ej. Capturas de pantalla o extractos de la línea de comandos.	Sí.

Para problemas y mensajes de error, se proporcionan los síntomas y las soluciones paso a paso.	Sí. En la documentación se ofrece un ítem que explica el procedimiento para reportar y solucionar errores.
No usa términos como "intuitivo", "fácil de usar", "fácil de usar", "simple" u "obviamente", a menos que sea parte de citas de usuarios satisfechos	Sí.
Indica los nombres y la sintaxis de los comandos, dice qué menú usar, enumera los parámetros y los mensajes de error exactamente como aparecen o deben escribirse.	Sí.
Documentación de la API, por ejemplo, JavaDoc o Doxygen, documentos API, por ejemplo. Archivos de configuración, nombres de propiedades, etc.	Sí. La documentación del API es generada por la librería <i>Loopback</i> .
Está guardada bajo una versión de control junto con el código.	No. Por ahora es solo un documento, pero se planea crear una página interactiva de documentación
Está en el sitio web del proyecto	No. La documentación se encuentra en un estado de desarrollo, en este proyecto se presenta lo que podría ser el documento final.
La documentación en el sitio web del proyecto deja en claro a qué versión del software se aplica la documentación.	No.

7.1.3 Capacidad de compilación. La capacidad de compilación determina qué tan fácil es cumplir los requisitos para construir el software y su capacidad de ser compilado en múltiples sistemas operativos. Siger muestra una alta capacidad de compilación ya que esta no solo es automatizada sino sus dependencias están correctamente organizadas dentro del proyecto y puede ser fácilmente instalado en cualquier servidor.

Tabla 5.

Tabla de criterios de capacidad de compilación.

Capacidad de compilación.	Sí/No, Justificación si es necesario
Existen instrucciones para construir el software.	Sí
Compilación automática usando un asistente de compilación.	Sí. El <i>front-end</i> usa <i>webpack</i> para su construcción en binarios para ser desplegada en la web, esta construcción es realizada automáticamente.
El proyecto lista todas las dependencias de terceros que no están empaquetados, junto con sus direcciones web, versiones adecuadas, y licencias si son necesarias.	Sí. En cada proyecto existe un archivo llamado <i>package.json</i> en donde se especifica las dependencias y versiones usadas. También las dependencias son instaladas en una carpeta llamada <i>node_modules</i> , en donde se incluye la información de cada dependencia y sus licencias.
Usa un software de administración de dependencias para descargar paquetes automáticamente.	Sí. Para los dos proyectos de <i>back-end</i> y <i>front-end</i> se usa el administrador de dependencias de node o NPM. Este software lee el archivo <i>package.json</i> para instalar las dependencias.
Todas las dependencias obligatorias de terceros están actualmente disponibles.	Sí.
Todas las dependencias opcionales de terceros están actualmente disponibles.	Sí.

7.1.4 Capacidad de instalación. La capacidad de instalación verifica qué tan fácil es instalar y desinstalar los módulos de la aplicación web en determinada plataforma. SIGER muestra una alta capacidad de instalación, pues ofrece un procedimiento estándar para hacerlo, y este puede ser usado en múltiples plataformas.

Tabla 6.

Tabla de criterios de capacidad de instalación.

Capacidad de instalación	Sí/No, Justificación si es necesario
Existen instrucciones para instalar el software.	Sí. En este proyecto se encuentran especificadas las instrucciones de instalación.
Los instaladores permiten al usuario especificar la ruta de instalación	Sí. Se puede clonar el repositorio en cualquier carpeta que se desee
Los desinstaladores eliminan todos los archivos o advierten al usuario de los archivos que no se eliminaron y dónde están.	Sí. La aplicación no tiene comando de eliminación, pero todos los archivos se generan dentro de la carpeta del proyecto, solo con borrarla del sistema queda desinstalada.

7.1.5 Curva de aprendizaje. La curva de aprendizaje evalúa qué tan fácil es aprender a realizar tareas simples y tareas complejas en la aplicación web, esto se hace basado en la documentación. SIGER demuestra un alto nivel de curva de aprendizaje, aunque se planea para futuras versiones de la documentación cubrir todos los casos de uso posibles.

Tabla 7.

Tabla de criterios de curva de aprendizaje.

Curva de aprendizaje	Sí/No, Justificación si es necesario
Se proporciona una guía de inicio que describe un ejemplo básico de uso del software.	Sí.
Se proporcionan instrucciones para muchos	Sí.

casos de uso básicos.

Se proporcionan instrucciones que apoyan todos los casos de uso.	No. La documentación está en desarrollo y aunque ofrece los casos de uso básicos no cubre todas las posibles actividades que se pueden realizar.
Se proporcionan guías de referencia para todas las opciones de línea de comandos, GUI y configuración.	Sí.
La documentación de la API se proporciona para los desarrolladores.	Sí. Aunque se encuentra en un estado inicial se puede observar los diferentes módulos del API y sus operaciones.

7.2 Sostenibilidad y mantenibilidad

Este criterio evalúa los derechos y licencia del software, su portabilidad o capacidad de ser desplegada y compilada en múltiples sistemas operativos. También evalúa las políticas creadas para introducir cambios y la calidad del código fuente. En el anexo A y B se puede encontrar un manual de instalación y los requerimientos mínimos necesarios para instalar SIGER.

7.2.1 Derechos de autor. Este ítem determina si el proyecto posee licencia de derechos de autor y qué tipo de licencia posee. SIGER posee una licencia MIT. Adicionalmente SIGER está en proceso de ser parte de la propiedad intelectual de la UIS, una solicitud para registro de software hecha el 1 de agosto de 2019, ver Anexo D.

Tabla 8.

Tabla de criterios para derechos de autor.

Derechos de autor	Sí/No, Justificación si es necesario
El proyecto tiene derechos de autor	Sí.
El proyecto dice quién o quienes desarrollaron el proyecto	Sí.
Cada archivo en el código fuente tiene una cabecera de licencia.	No.
El proyecto tiene una licencia de código abierto	Sí

7.2.2 Accesibilidad. La accesibilidad mide el nivel de acceso que tiene aplicación web, tanto su código fuente como el acceso a través de otras plataformas. SIGER está disponible en el enlace siger.uis.edu.co y se puede ingresar desde cualquier dispositivo, para su código fuente está guardando en un repositorio en línea y puede ser accedido también solo si se tiene acceso.

Tabla 9.

Tabla de criterios para accesibilidad.

Accesibilidad	Sí/No, Justificación si es necesario
La aplicación se puede acceder gratuitamente	Sí. Se puede acceder en siger.uis.edu.co
Las distribuciones de fuentes de código están disponibles (ya sea de forma gratuita, de pago, de registro).	Sí. Si tiene acceso al repositorio Bitbucket y permisos de lectura.
Las distribuciones de fuentes de código están disponibles sin la necesidad de ningún registro o autorización de acceso por parte del proyecto.	No. El repositorio Bitbucket es privado y necesita permisos para poder leer o descargar el código.
Posibilidad de navegar por el repositorio de código fuente en línea, desde un navegador.	Sí. Si tiene permisos en Bitbucket
El repositorio se aloja externamente en una única organización / institución en un repositorio sostenible de terceros (por ejemplo, SourceForge, GoogleCode, LaunchPad,	Sí, está guardado en Bitbucket.

GitHub) que vivirá más allá de la vida útil de cualquier línea de financiamiento actual.

7.2.3 Portabilidad. Este ítem evalúa la capacidad de la aplicación de ser desplegada y accesada en distintas plataformas. SIGER puede ser desplegada en cualquier tipo de servidor y accedida desde todos los navegadores.

Tabla 10.
Tabla de criterios para portabilidad.

Portabilidad	Sí/No, Justificación si es necesario
La aplicación puede ser construida y desplegada en Windows 10.	Sí.
La aplicación puede ser construida y desplegada en Windows 7.	Sí, si el navegador lo soporta.
La aplicación puede ser construida y desplegada en UNIX/Linux.	Sí
La aplicación puede ser construida y desplegada en Solaris.	Sí
La aplicación puede ser construida y desplegada en RedHat.	Sí
La aplicación puede ser construida y desplegada en Debian.	Sí
La aplicación puede ser construida y desplegada en Fedora.	Sí
La aplicación puede ser construida y desplegada en Ubuntu.	Sí
La aplicación puede ser construida y desplegada en MacOSX.	Sí
La aplicación corre en Internet Explorer	Sí
La aplicación corre en Mozilla Firefox	Sí

La aplicación corre en Google Chrome	Sí
La aplicación corre en Opera	Sí
La aplicación corre en Safari	Sí

7.2.4 Capacidad de soporte. Este ítem evalúa la calidad de soporte ofrecido por la aplicación web. SIGER ofrece una opción de enviar correos con problemas presentados, actualmente la mayoría de los errores son presentados en las reuniones de consejo. Se planea agregar una opción de soporte en la página web para recibir retroalimentación y reporte de errores desde la web.

Tabla 11.

Tabla de criterios para capacidad de soporte.

Capacidad de soporte	Sí/No, Justificación si es necesario
La aplicación ofrece una opción de soporte.	No.
La documentación describe cómo obtener soporte	Sí.
El proyecto tiene un correo específico	Sí
Existe un blog y es regularmente usado	No. Pero va a estar disponible en futuras versiones.

7.2.5. Analizabilidad. La analizabilidad evalúa la calidad del código fuente de la aplicación y cómo esta se relaciona con su diseño y arquitectura inicial. SIGER es un proyecto en desarrollo activo, realizado con la metodología SCRUM por lo cual no entrega un producto completo libre de errores sino se actualiza semanalmente agregando nuevas características y haciendo mejoras y limpieza en el código, aún así muestra un nivel alto de analizabilidad pues está programada bajo estándares de código y comentado en caso de ser necesario.

Tabla 12.

Tabla de criterios para analizabilidad.

Analizabilidad	Sí/No, Justificación si es necesario
El código fuente está estructurado en módulos o paquetes	Sí
La estructura del código fuente se relaciona claramente con la arquitectura o el diseño.	Sí
Se proporcionan archivos de proyecto para IDE.	No. El proyecto puede ser desarrollado en cualquier IDE.
El repositorio de código fuente es un sistema de control de revisiones.	Sí
Los comentarios del código fuente están escritos en un lenguaje de marcado de generación de documentos API, p. Ej. JavaDoc o Doxygen.	No. Pero estará en futuras versiones
El código fuente está diseñado e indentado bien.	Sí.
El código fuente utiliza nombres sensibles de clase, paquete y variables.	Sí.
No hay archivos de código fuente antiguos que deban ser manejados por el control de versiones, por ejemplo. "SomeComponentOld.java".	Sí.
No hay código comentado.	No, existe código comentado pero será removido en futuras versiones

No hay TODOs en el código.

No, todavía existen algunos TODO para optimizar partes del código, serán removidos en futuras versiones

El proyecto recomienda y usa estándares de programación. Sí.

8. Conclusiones

Se diseñó e implementó una aplicación web llamada Sistema de Gestión de Reuniones, SIGER (siger.uis.edu.co), con el propósito de mejorar la gestión de la información de reuniones de los Consejos del Instituto de Proyección Regional y Educación a Distancia (IPRED). Desde el punto de vista de los usuarios, la implementación de la aplicación web SIGER ha facilitado los procesos de gestión de información de las reuniones del Consejo de Programas y de Instituto del IPRED, lo cual corresponde a un aporte significativo en la resolución de una necesidad de la Universidad y corresponde con un aporte destacado de aplicación de técnicas foráneas (Gobierno en Línea) adaptadas a las condiciones del país. Es de destacar que la aplicación SIGER está en uso intensivo en la Universidad, por parte del IPRED, el cual trata más de veinte casos por semana utilizando la aplicación. En este sentido, se concluye que la aplicación cumple con los objetivos para la cual fue creada, pues permite organizar la información generada por los consejos y mejora su distribución entre las diferentes entidades de la UIS, para eso se apoya en su arquitectura modular, la cual permite que la aplicación pueda ser desplegada en cualquier servidor independientemente del sistema operativo, desde el punto de vista del usuario administrador esta característica hace la aplicación más confiable y ofrece una mejor mantenibilidad, reduciendo la complejidad en la configuración y permitiendo simplificar el despliegue de nuevos cambios. Adicionalmente, expone sus datos mediante una arquitectura REST cuyo protocolo de acceso está basado en *tokens* y roles de acceso, esto permite restringir accesos no autorizados y gracias a la arquitectura REST permite acceso desde varios clientes (navito, web, móviles), haciendo de la aplicación un repositorio centralizado de datos que puede ser usado por otras unidades de la UIS para obtener los últimos cambios en las actas y temas, aumentando la accesibilidad de los datos generados, esta

característica redujo el tiempo en el cual son distribuidas las respuestas generadas con por los consejos del IPRED que actualmente usan SIGER.

Referencias Bibliográficas

- [1] Baró, Luis Fernando, & Gómez, Ricardo De la infraestructura a la apropiación social: panorama sobre las políticas de las tecnologías de información y comunicación (tic) en Colombia
- [2] Boicea, A., Radulescu, F., & Agapin, L. (2012). MongoDB vs Oracle -- Database Comparison. 2012 Third International Conference on Emerging Intelligent Data And Web Technologies. doi: 10.1109/eidwt.2012.32
- [3] Colrain M. Zuppo. Defining ICT in a Boundaryless Word: The Development of a Working Hierarchy.
- [4] Fielding, R. (2000). Fielding Dissertation: CHAPTER 5: Representational State Transfer (REST). Retrieved from https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- [5] Hans J Schnoll. E-Government: Information, Technology, and Transformation: Information.
- [6] Jackson, M., Crouch, S., & Baxter, R. Software Evaluation Guide | Software Sustainability Institute. Retrieved from <https://www.software.ac.uk/resources/guides-everything/software-evaluation-guide>
- [7] Líppez De Castro, S., & García Alonso, R. (2016). Ciudadanos y gobierno electrónico: la orientación al ciudadano de los sitios Web municipales en Colombia para la promoción de la participación. Universitas Humanística, 82(82). doi: 10.11144/javeriana.uh82.cgeo
- [8] Mete Yildiz. E-Government Research: Reviewing the Literature, Limitations, and Ways Forward.

- [9] Njoh, A. (2018). The relationship between modern Information and Communications Technologies (ICTs) and development in Africa. *Utilities Policy*, 50, 83-90. doi: 10.1016/j.jup.2017.10.005
- [10] Noor, Marhaini Modh (2018). Rural Community Digital Technology Connectedness: Does ICT in Rural Area Contributes to Rural Development in Malaysia?
- [11] Park, C. (2017). A study on effect of entrepreneurship on entrepreneurial intention. *Asia Pacific Journal of Innovation and Entrepreneurship*, 11(2), 159-170. doi: 10.1108/apjie-08-2017-024
- [12] Schwaber, K., & Sutherland, J. Scrum Guide | Scrum Guides. Retrieved from <https://www.scrumguides.org/scrum-guide.html>
- [13] S.C.J. Palvia and Sushil S. Sharma / E-Government and E-Governance: Definitions/Domain
- [14] Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to Build High-Performance Network Programs. *IEEE Internet Computing*, 14(6), 80-83. doi: 10.1109/mic.2010.145
- [15] UNESCO UIS. (2019). Information and communication technologies (ICT) | Retrieved from <http://uis.unesco.org/en/glossary-term/information-and-communication-technologies-ict>

Apéndices

Apéndice A. Descripción del hardware y software requeridos para el despliegue.

La aplicación web SIGER necesita el montaje de dos módulos para su despliegue. El primero es el *back-end*, que consiste en un proyecto escrito en *Javascript* y funcionará como interfaz entre la base de datos MongoDB y el cliente para el manejo de datos. El *back-end* montará un API REST que permitirá el manejo de datos. El API REST de SIGER podrá ser desplegado en un servidor con cualquier sistema operativo, pero es recomendado sistemas operativos basados en Linux, pues ofrecen mayor versatilidad. El segundo módulo es el *front-end*, también escrito en *Javascript*, y consistirá en un servicio que enviará una copia de la aplicación web al cliente para que este administre los datos. El *front-end* llamará al *back-end* internamente para realizar las operaciones de administración, como creación, edición, búsqueda, exportación de actas, entre otros. Lo anterior significa que todas las operaciones de datos se harán desde el servidor. Para el despliegue inicial de SIGER se usará un servidor Azure con las siguientes especificaciones:

- Ubuntu Server 18.04.2 LTS (Mínimo Ubuntu 14.04).
- Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz (Mínimo Intel Atom processor @ 1.6 GHz).
- 4GB de RAM (Mínimo 2GB RAM).
- 60GB de almacenamiento (Mínimo 40GB de almacenamiento).

El requerimiento más relevante es el almacenamiento, pues SIGER funciona como un servidor de archivos y necesita suficiente memoria para los adjuntos que van con cada tema tratado en las

reuniones. El servidor también deberá poseer firewall para permitir abrir puertos y ser accedido desde el exterior y soportar conexiones por SSH.

Para su correcto funcionamiento SIGER se apoya se las siguientes herramientas:

- **MongoDB:** Motor de base de datos NoSQL. Recomendado la instalación de la última versión.
- **NodeJS:** Entorno de ejecución *Javascript* para servidores. En Node se ejecutarán el *back-end* y *front-end* de SIGER. Recomendado la instalación de la versión 8 LTS.
- **NPM:** Es un manejador de paquetes para NodeJS, es necesario para desplegar los módulos a partir del código fuente.
- **Nginx:** Servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico. Recomendado versión que viene por defecto en Ubuntu. Necesario para redirección de puertos de los servicios Node hacia el puerto 22.
- **PM2:** Es un administrador de procesos de larga duración en Node, cada proceso se le hace seguimiento con estadísticas.
- **Serve:** Librería de Node.JS usada para servir archivos estáticos, útil para la descarga de archivos adjuntos de actas, y descarga de archivos del front-end para su ejecución en el cliente.
- **Git:** Es importante tener instalado un cliente de Git, ya que el código fuente del back-end y front-end de SIGER estarán en un repositorio. Este repositorio se

clonará en el servidor y se desplegará con uso de comandos y las herramientas PM2 y Serve.

- **Certbot:** Certificado SSL/TLS gratuito, útil para habilitar navegación encriptada.

Apéndice B. Descripción de la instalación.

Para instalar y desplegar el back-end y front-end de SIGER se deben realizar los siguientes pasos, los pasos fueron ejecutados en el sistema operativo recomendado para SIGER, Ubuntu Server 18.04.2.

Actualización del servidor.

Es necesario actualizar el repositorio de paquetes del servidor previo a la instalación de SIGER para obtener siempre las últimas versiones de las herramientas necesarias. Para ello se ejecutan los siguientes comandos:

```
sudo apt-get update && sudo apt-get upgrade  
sudo apt-get dist-upgrade  
sudo reboot
```

Con estos comandos se actualizará no solo los paquetes de Ubuntu sino también la versión del sistema operativo. El comando de reinicio es opcional.

Instalación de MongoDB

La instalación de MongoDB se hace siguiendo el manual oficial para la versión Ubuntu Server 18.04.2, puede cambiar dependiendo de la versión de Ubuntu que se use, pero en general es importante instalar la última versión disponible, actualmente la 3.6, para ello se ejecutan los siguientes comandos:

- Comando para agregar llaves de seguridad para prevenir ataques DDOS.

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv  
2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
```

- Comando para indexar los instaladores al manejador de paquetes de Ubuntu.

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu  
xenial/mongodb-org/3.6 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-  
org-3.6.list
```

- Comando para actualizar el manejador de paquetes con los nuevos instaladores.

```
sudo apt-get update
```

- Comando para instalar MongoDB en Ubuntu.

```
sudo apt-get install -y mongodb-org
```

- Comando para iniciar el servicio de MongoDB en Ubuntu.

```
sudo service mongod start
```

Una vez realizada la instalación se podrá probar si fue exitosa ejecutando el comando “mongo” en la consola, esto es así cuando se entra a la interfaz de aplicación de MongoDB sin problemas.

Instalación de NodeJS

Node es el entorno de aplicaciones y servicios para servidor necesario para ejecutar los módulos de *back-end* y *front-end*. Se recomienda instalar la versión 8.0 LTS. Para su instalación se deben ejecutar los siguientes comandos:

- Comando para agregar los instaladores al manejador de paquetes de Ubuntu.

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
```

- Comando para instalar la versión 8 de NodeJS en Ubuntu

```
sudo apt-get install -y nodejs
```

- Comando para instalar el paquete build-essential necesario para algunas librerías de Node.

```
sudo apt-get install -y build-essential
```

Instalación de Nginx

Nginx es necesario para desplegar SIGER en la web y poder acceder a la aplicación a través de una *url* o *endpoint*. Para instalar se debe ejecutar “sudo apt-get install nginx”. El comando de instalación iniciará el servicio Nginx automáticamente.

Configuración de Firewall

Para reducir riesgos potenciales de seguridad en el servidor es necesario descargar y activar un Firewall que bloquee el acceso a puertos, para Ubuntu existe un *ufw*, para configurar y permitir solo puertos autorizados por Nginx y el puerto necesario para conexión por SSH se ejecutan los siguientes comandos:

- Comandos para agregar reglas SSH y Nginx

```
sudo ufw allow ssh
```

```
sudo ufw allow 'Nginx Full'
```

- Comando para activar el firewall

```
sudo ufw enable
```

Instalación de PM2 y Serve

PM2 y Serve son dos librerías de NodeJS para manejar procesos Node y servir archivos estáticos a la web respectivamente, su instalación de manera global se hace ejecutando el comando “sudo npm install pm2 serve -g”

Clonación de repositorios

SIGER está disponible en Bitbucket, pero el repositorio no es público pues es solo accesible para miembros del IPRED. Para requerir acceso al repositorio se debe enviar un correo a maceveva@uis.edu.co con el asunto “SIGER - ACCESO A REPOSITORIO”. Se le responderá con los pasos necesario para obtener acceso. Este correo proporciona al final una llave y un enlace para clonar el repositorio con el comando *clone*.

Inicio de servicios Node

Una vez clonados los repositorios en Node el siguiente paso es iniciar los servicios Node del *back-end* y el *front-end*, para esto se usará PM2. El proceso de *back-end* se ejecutará en el puerto 3000 y el proceso de *front-end* en el 3001.

- Comandos para iniciar el back-end (Dentro de la carpeta del proyecto)

```
sudo npm install
```

```
pm2 start . --name “siger-backend”
```

- Comandos para iniciar el front-end (Dentro de la carpeta del proyecto)

```
pm2 start serve --name "siger-frontend" -- -s -p 3001 build
```

Configuración de Nginx y HTTPS

Hasta este punto el servidor de SIGER tendrá iniciados los procesos Node correspondientes al *back-end* y *front-end*, pero no se pueden acceder desde los clientes, para esto es necesario configurar Nginx. La configuración de Nginx se guarda en el archivo ubicado en el directorio `/etc/nginx/sites-available/default` en los servidores basados en Ubuntu. La configuración hará una redirección de puertos de los procesos Node de *back-end* y *front-end* al puerto 80 de HTTP y el puerto 443 de HTTPS. Para el certificado HTTPS se usará Certbot y SIGER será servido en la URL `siger.uis.edu.co`.

La instalación de Certbot se hace con los siguientes pasos.

```
sudo add-apt-repository ppa:certbot/certbot
sudo apt-get update
sudo apt-get install python-certbot-nginx
sudo certbot --nginx -d siger.uis.edu.co
```

Una vez instalado Certbot se sobrescribe la siguiente configuración al archivo *default*:

```
upstream api_node_js {
    server 127.0.0.1:3000;
}

server {
```

```
if ($host = siger.uis.edu.co) {  
    return 301 https://$host$request_uri;  
}  
  
listen 80 default_server;  
  
listen [::]:80 default_server;  
  
server_name siger.uis.edu.co;  
  
return 301 https://siger.uis.edu.co$request_uri;  
}  
  
server {  
  
    listen [::]:443 ssl ipv6only=on; # managed by Certbot  
  
    listen 443 ssl; # managed by Certbot  
  
    root /var/www/html;  
  
    # Add index.php to the list if you are using PHP  
  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name siger.uis.edu.co www.siger.uis.edu.co;  
  
    location / {  
  
        proxy_pass http://127.0.0.1:3001;  
  
    }  
}
```

```
location /api {  
  
    proxy_set_header X-Real-IP $remote_addr;  
  
    proxy_set_header X-Forwarded-  
  
    For                                $proxy_add_x_for  
  
    warded_for;  
  
    proxy_set_header Host $http_host;  
  
    proxy_set_header X-NginX-Proxy true;  
  
  
    proxy_pass http://api_node_js;  
  
    proxy_redirect off;  
  
    }  
  
    ssl_certificate /etc/letsencrypt/live/siger.uis.edu.co/fullchain.pem;  
  
    ssl_certificate_key /etc/letsencrypt/live/siger.uis.edu.co/privkey.pem;  
  
    }
```

Después de escribir esta configuración se reinicia Nginx usando el comando “nginx -s reload”. Para probar si la configuración exitosa de ingresar a la URL siger.uis.edu.co, debe mostrar la vista de consulta de SIGER (Figura 22).

Sistema de gestión de reuniones INGRESAR

Actas BUSCAR ACTUALIZAR

Año
2019-01-01

NÚMERO DE ACTA	FECHA DE INICIO	FECHA DE FINALIZACIÓN	PDF	
14	5/8/2019, 7:00:00 AM	5/8/2019, 9:00:00 AM		MOSTRAR
13	4/26/2019, 8:00:00 AM	4/23/2019, 10:00:00 AM		MOSTRAR
12	4/12/2019, 10:00:00 AM	4/11/2019, 12:00:00 PM		MOSTRAR
11	4/2/2019, 8:00:00 AM	4/2/2019, 10:00:00 AM		MOSTRAR
10	3/28/2019, 10:00:00 AM	3/28/2019, 12:00:00 AM		MOSTRAR
09	3/21/2019, 10:00:00 AM	3/21/2019, 12:00:00 PM		MOSTRAR
08	3/12/2019, 8:00:00 AM	3/12/2019, 10:00:00 AM		MOSTRAR
07	3/5/2019, 8:00:00 AM	3/5/2019, 10:00:00 AM	Acta N° 7.pdf	MOSTRAR
06	2/26/2019, 8:00:00 AM	2/26/2019, 10:00:00 AM		MOSTRAR
05	2/19/2019, 8:00:00 AM	2/19/2019, 10:00:00 AM		MOSTRAR

1-10 de 14 1 2 SIGUIENTE >

Figura 22. Vista de consulta. siger.uis.edu.co.

Apéndice C. Manual de Usuario.

SIGER o Sistema de Gestión de Reuniones es una aplicación web que administra las actas generadas en las reuniones de concejo del IPRED, permitiendo eliminar los reprocesos que ocurren en su creación. Esta herramienta fue creada para las secretarías, coordinadores y estudiantes del IPRED, y su diseño está basado en tres pilares fundamentales. El primero es la accesibilidad, pues soporta navegación de cualquier dispositivo y presenta la información de forma clara y concisa. Después se encuentra la retroalimentación, pues está en constante desarrollo y se ajusta a las necesidades de sus usuarios. Y finalmente la interconexión, pues es una herramienta que permite la distribución de información de actas entre los consejos del IPRED y entre sus diferentes unidades. Este manual ofrecerá una introducción al manejo de SIGER y de cómo administrar la información de actas.

Solicitud de usuario

Antes de realizar cualquier operación en SIGER, es necesario hacer una solicitud para obtener un usuario en la aplicación. Esta solicitud se puede hacer a la Ingeniera Maria Alejandra Acevedo por el correo maceveva@uis.edu.co, el cuerpo del correo debe contener las credenciales de ingreso que el cliente desee, el correo y una contraseña provisional. En las próximas veinticuatro (24) horas el cliente recibirá confirmación de la creación de su usuario y un enlace para entrar a SIGER vía web. Se recomienda al usuario cambiar la contraseña una vez ingrese a la aplicación.

Ingreso y Consulta

Una vez el cliente tenga su usuario creado, puede iniciar sesión ingresando a la dirección web siger.uis.edu.co. Se presentará la vista de consulta (Figura 23), en esta pantalla el cliente puede buscar un acta específica por palabra clave o por número de acta. Se puede organizar la lista de actas por número y por fecha de creación. Si está disponible se puede descargar el acta en PDF. En cada entrada de la lista hay un botón de “Mostrar”, al presionarlo se muestra el detalle del acta y los temas que trata.

En la esquina superior hay un botón de “Ingresar”, al presionarlo se muestra la vista de ingreso (Figura 24), en donde el cliente podrá ingresar sus credenciales para entrar a la aplicación web.

The screenshot displays the 'Sistema de gestión de reuniones' interface. At the top right, there is a red 'INGRESAR' button. Below it, a search bar contains 'Mostrar filtros de búsqueda' and 'Año: 2019-01-01'. The main content is a table with the following columns: 'NÚMERO DE ACTA', 'FECHA DE INICIO', 'FECHA DE FINALIZACIÓN', and 'PDF'. The table lists 10 meeting minutes. Each row has a 'Mostrar' button. A red box highlights the 'Mostrar' button for act 14, with the label 'Mostrar detalle de acta'. Another red box highlights the 'Acta N° 7.pdf' link in the PDF column for act 07, with the label 'Descargar acta'. The bottom of the page shows pagination: '1-10 de 14', '1', '2', and 'SIGUIENTE >'.

NÚMERO DE ACTA	FECHA DE INICIO	FECHA DE FINALIZACIÓN	PDF
14	5/8/2019, 7:00:00 AM	5/8/2019, 9:00:00 AM	Mostrar
13	4/26/2019, 8:00:00 AM	4/23/2019, 10:00:00 AM	Mostrar
12	4/12/2019, 10:00:00 AM	4/11/2019, 12:00:00 PM	Mostrar
11	4/2/2019, 8:00:00 AM	4/2/2019, 10:00:00 AM	Mostrar
10	3/28/2019, 10:00:00 AM	3/28/2019, 12:00:00 AM	Mostrar
09	3/21/2019, 10:00:00 AM	3/21/2019, 12:00:00 PM	Mostrar
08	3/12/2019, 8:00:00 AM	3/12/2019, 10:00:00 AM	Mostrar
07	3/5/2019, 8:00:00 AM	3/5/2019, 10:00:00 AM	Acta N° 7.pdf
06	2/26/2019, 8:00:00 AM	2/26/2019, 10:00:00 AM	Mostrar
05	2/19/2019, 8:00:00 AM	2/19/2019, 10:00:00 AM	Mostrar

Figura 23. Vista de consulta.

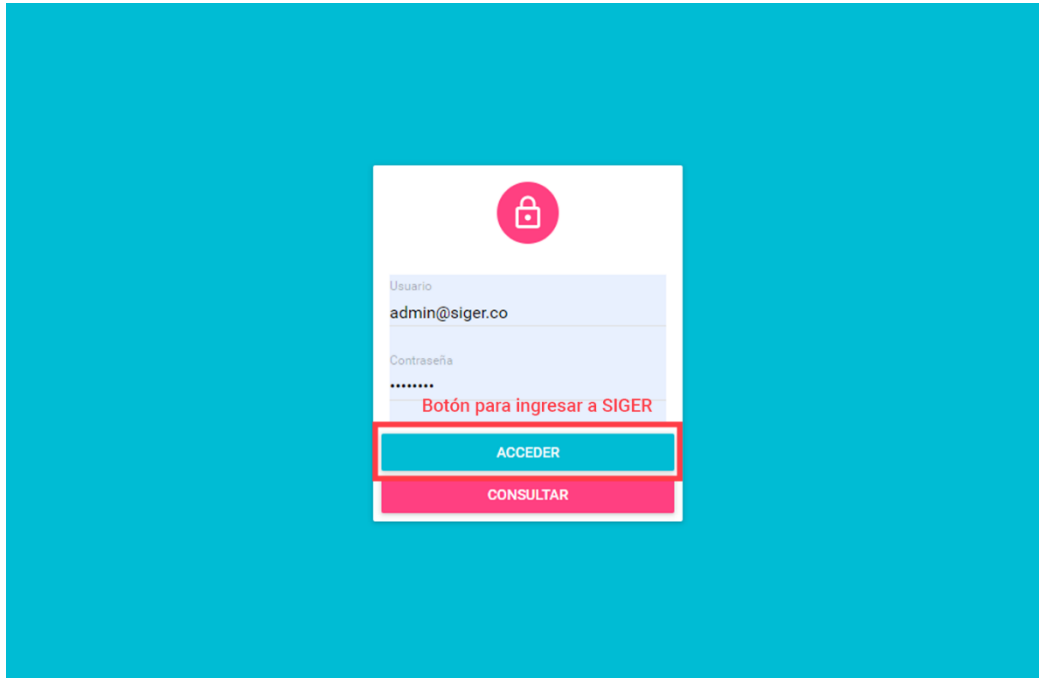


Figura 24. Vista de ingreso.

Vista general de SIGER

SIGER está compuesto de tres partes principales (Figura 25). La primera parte es el menú, en el que se muestra la lista de recursos disponibles para administrar. El segundo es la vista de detalle, esta vista puede cambiar dependiendo de la acción que se esté realizando, por defecto muestra la acción de listar, en donde se listan los datos de un recurso. La tercera es la barra de menú, desde la cual se puede cambiar de consejo, si el cliente pertenece a más de uno. El recurso seleccionado por defecto es el de “Actas”.

The screenshot displays the 'Sistema de gestión de reuniones - Consejo de Programas' interface. On the left is a sidebar menu with options like 'Actas', 'Temas', 'Estados de temas', 'Tipos de temas', 'Tipos de tema de estudiante', 'Tipos de condicionalidad', 'Programas', 'Consejos', 'Sedes', 'Tipos de comentarios', 'Usuarios', and 'Cerrar sesión'. The main area shows a table of minutes with columns for 'NÚMERO DE ACTA', 'FECHA DE INICIO', 'FECHA DE FINALIZACIÓN', and 'PDF'. The table contains 14 rows of data. At the bottom, there is a pagination control showing '1-10 de 14' and 'SIGUIENTE >'.

NÚMERO DE ACTA	FECHA DE INICIO	FECHA DE FINALIZACIÓN	PDF	ACTIONS
14	5/8/2019, 7:00:00 AM	5/8/2019, 9:00:00 AM		EDITAR
13	4/26/2019, 8:00:00 AM	4/23/2019, 10:00:00 AM		EDITAR
12	4/12/2019, 10:00:00 AM	4/11/2019, 12:00:00 PM		EDITAR
11	4/2/2019, 8:00:00 AM	4/2/2019, 10:00:00 AM		EDITAR
10	3/28/2019, 10:00:00 AM	3/28/2019, 12:00:00 AM		EDITAR
09	3/21/2019, 10:00:00 AM	3/21/2019, 12:00:00 PM		EDITAR
08	3/12/2019, 8:00:00 AM	3/12/2019, 10:00:00 AM		EDITAR
07	3/5/2019, 8:00:00 AM	3/5/2019, 10:00:00 AM	Acta N° 7.pdf	EDITAR
06	2/26/2019, 8:00:00 AM	2/26/2019, 10:00:00 AM		EDITAR
05	2/19/2019, 8:00:00 AM	2/19/2019, 10:00:00 AM		EDITAR

Figura 25. Vista general de SIGER.

Lista de Recursos

La vista de lista de recursos (Figura 26) muestra los datos creados para determinado recurso. Los encabezados de la lista pueden ser presionados para ordenarla por el campo seleccionado. Esta vista posee una barra de acciones desde la cual se puede crear un nuevo dato o buscar un dato específico. La vista de lista cambia dependiendo del recurso, se mostrarán los campos relevantes para su búsqueda, en el caso de el recurso de “Actas” estos son: número de acta, fecha de inicio, fecha de realización.

Actas

Barra de acciones

BUSCAR + CREAR ACTUALIZAR

Año: 2019-01-01

Encabezado de la lista.

NÚMERO DE ACTA	FECHA DE INICIO	FECHA DE FINALIZACIÓN	PDF
14	5/8/2019, 7:00:00 AM	5/8/2019, 9:00:00 AM	EDITAR
13	4/26/2019, 8:00:00 AM	4/23/2019, 10:00:00 AM	EDITAR
12	4/12/2019, 10:00:00 AM	4/11/2019, 12:00:00 PM	EDITAR
11	4/2/2019, 8:00:00 AM	4/2/2019, 10:00:00 AM	EDITAR
10	3/28/2019, 10:00:00 AM	3/28/2019, 12:00:00 AM	EDITAR
09	3/21/2019, 10:00:00 AM	3/21/2019, 12:00:00 PM	EDITAR
08	3/12/2019, 8:00:00 AM	3/12/2019, 10:00:00 AM	EDITAR
07	3/5/2019, 8:00:00 AM	3/5/2019, 10:00:00 AM	Acta N' 7.pdf EDITAR
06	2/26/2019, 8:00:00 AM	2/26/2019, 10:00:00 AM	EDITAR
05	2/19/2019, 8:00:00 AM	2/19/2019, 10:00:00 AM	EDITAR

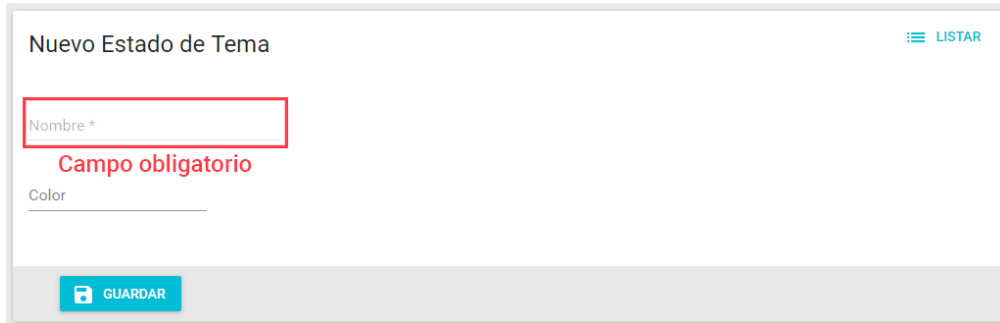
1-10 de 14

1 2 SIGUIENTE >

Figura 26. Vista de Lista de Recursos.

Vista de Creación de Recursos

Para acceder a esta vista se presiona el botón de “Crear” en la barra de acciones de la vista de listar recursos. En esta vista (Figura 27) se observa un formulario con los campos necesarios para crear un nuevo dato. Los campos con un asterisco (*) serán obligatorios. Para guardar el dato se redirecciona a la vista de lista para verificar que el dato fue creado exitosamente.



Nuevo Estado de Tema LISTAR

Nombre *
Campo obligatorio

Color

GUARDAR

Figura 27. Vista de creación para recurso “Estado de Tema”.

Vista de Edición de Recursos

En la vista de lista de recursos, al final de cada fila de los datos listados, se encuentra un botón de “Editar”, al presionarlo se muestra la vista de edición de recursos (Figura 28). Esta vista es similar a la vista de creación pues posee los mismos campos del formulario. En ocasiones cambia la barra de acciones y aparecen nuevas como exportar o remover el dato.



Editar estado de tema: "No Favorable" LISTAR ELIMINAR ACTUALIZAR

Id
5b7dcf757c17546e6a6d9c31

Creado en
8/22/2018, 4:02:45 PM

Nombre *
No Favorable

Color
#FF5252

GUARDAR

Figura 28. Vista de edición de recursos.

Creación y edición de temas

La creación y edición de temas constituyen una parte fundamental de SIGER, pues el tema es una estructura en la que se expresa una situación o propuesta que debe ser resuelta, la evaluación de los temas es realizada por los coordinadores. Los temas poseen varias categorías y la vista de creación y edición cambia dependiendo de la categoría. La forma general (Figura 29) de un tema posee información sobre el acta a la cual está destinado, un título expresando la situación o propuesta y una descripción o justificación junto con una serie de archivos de soporte. Existe un caso especial, el caso de estudiante (Figura 30), el cual al ser evaluado se muestra en una tabla con la información pertinente, como el código, programa, nivel, condicionalidad del estudiante, entre otros.

Editar tema

EXPORTAR LISTAR ELIMINAR ACTUALIZAR

INFORMACIÓN CONCEPTOS CONCEPTOS COORDINADORES

Creado en
5/7/2019, 8:01:38 PM

¿Público?

¿Editable?

Tipo
Profesores

Título *
Convocatoria banco de elegibles Tutores 2019

Estado
En espera

Consejo
Consejo de Programas

Figura 29. Vista de edición de tema.

The screenshot shows a web interface for editing a topic. At the top, there are navigation buttons: EXPORTAR, LISTAR, ELIMINAR, and ACTUALIZAR. Below these is a section titled 'INFORMACIÓN'. The form contains the following fields:

- Creado en: 5/7/2019, 5:12:17 PM
- ¿Público?:
- ¿Editable?:
- Tipo: Estudiantes (Nuevo)
- Consejo: Consejo de Programas
- Estado: En espera
- Código *: 2187019
- Tipo de caso: Cancelación de Asignatura
- Sede: Bucaramanga

Figura 30. Vista de edición de tema de estudiantes.

Creación de Acta

El proceso y vista de creación de acta (Figura 31) es diferente a la vista de creación de otros recursos ya que en el acta está reunida toda la información correspondiente a los temas tratados en la reunión. La vista posee cuatro (4) pestañas cada una conteniendo información del acta. En la primera se determina el número de acta, la fecha en que se dio la reunión de consejo y el lugar. Si el acta está siendo creada para consulta se debe agregar también el archivo en PDF. La segunda y tercera pestaña corresponde a los integrantes de la reunión de consejo, por un lado se encuentra el quórum cuyos campos consisten en nombre y cargo. Por otro lado la lista de invitados cada uno con la justificación de su presencia en la reunión. La última pestaña muestra las actividades que se realizarán en la reunión, aparecen enumeradas en pestañas numéricas, se puede ver el nombre de

la actividad al pasar el cursor sobre el número. Dentro de cada pestaña se pueden agregar los temas que se van a evaluar. Una vez llenados los campos se puede guardar el acta.

Nueva acta LISTAR

INFORMACIÓN BÁSICA QUORUM INVITADOS ACTIVIDADES

Número de acta *

Fecha de inicio Time(hours, mins.) ✕

Fecha de finalización Time(hours, mins.) ✕

Consejo
Consejo de Programas

Lugar
Sala de Juntas de la Dirección

Acta escaneada
Sulte el archivo para cargar o haga click para seleccionar uno.

GUARDAR

Figura 31. Vista de creación de acta y sus pestañas.

Edición de Acta

La vista de edición de acta es una de las más importantes en SIGER, pues en ella se realizan acciones durante la reunión de consejo. Desde esta vista el cliente puede realizar las siguientes acciones:

- **Exportar acta:** En la barra de acciones se presiona el botón de “Exportar”. El formato del documento es Microsoft Word.

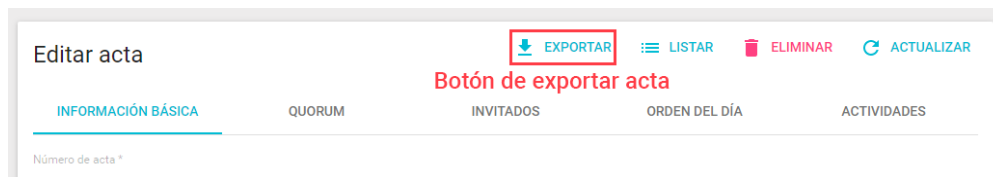


Figura 32. Botón de exportar acta.

- **Confirmar asistencia de quórum e invitados:** Para cada uno de los miembros agregados aparece un campo para confirmar asistencia. Esta información se ve reflejada en el acta exportada.

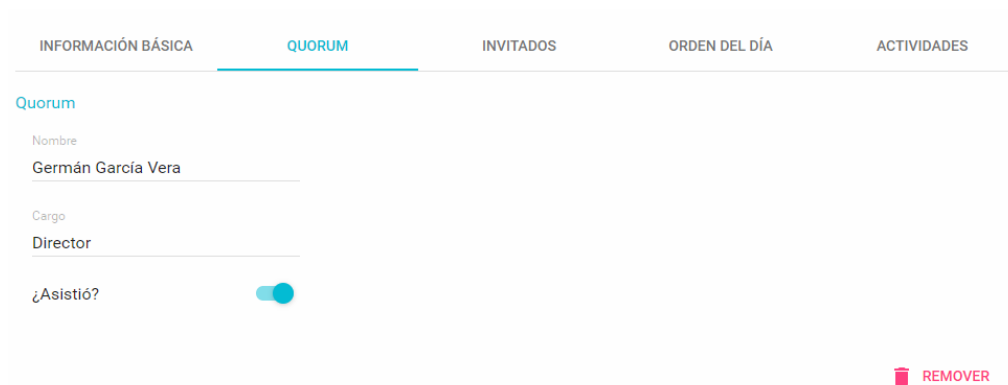


Figura 33. Confirmar asistencia en pestaña de Quórum.

- **Ajustar orden del día:** Se puede cambiar el orden de las actividades arrastrándose hacia arriba o hacia abajo.



Figura 34. Pestaña de orden del día. Arrastrando actividad para cambiar el orden.

- **Ver detalle de temas:** En los temas agregados en cada categoría, al presionar en el tema aparecerá una ventana con su detalle. Desde esta ventana se puede editar el tema y exportarlo. Existe un botón de guardar el tema y mostrar el siguiente o anterior.
- **Conceptuar rápidamente temas de estudiantes:** Los temas de estudiantes poseen una estructura similar y por lo general comprenden la mayoría de los temas tratados en una reunión. Existe una vista que permite observar y conceptuar temas de estudiantes de forma rápida. Para acceder a esta vista se presiona el botón “Mostrar tabla” en la pestaña.

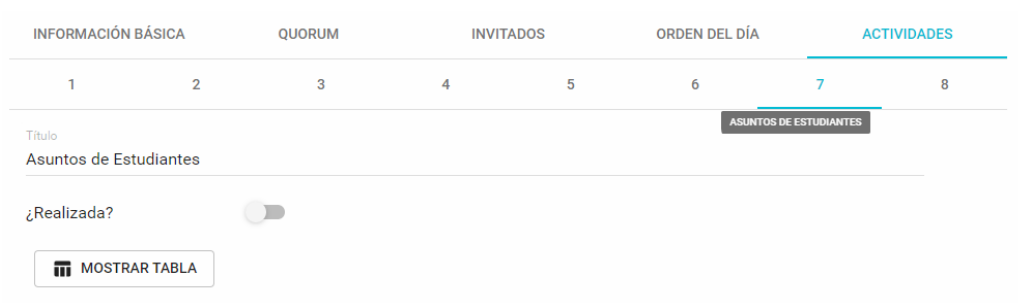


Figura 35. Botón de mostrar tablas. Pestaña de actividades, subpestaña de estudiantes.

Observaciones y Soporte

SIGER es una aplicación web diseñada para ser ajustada a las necesidades del IPRED, por lo cual estará en desarrollo activo y ofrecerá soporte a errores en actualizaciones hechas semanalmente.

Si la aplicación presenta algún problema o desea hacer retroalimentación lo puede hacer escribiendo por correo a la ingeniera Maria Alejandra Acevedo a maceveva@uis.edu.co.