Diseño de un modelo centralizado de autorizaciones para mejorar la seguridad en aplicaciones desarrolladas a la medida

Paula Andrea Cardona Ochoa

Instituto Tecnológico Metropolitano

Facultad de Ingeniería

Medellín, Colombia

2021

Diseño de un modelo centralizado de autorizaciones para mejorar la seguridad en aplicaciones desarrolladas a la medida

Paula Andrea Cardona Ochoa

Tesis o trabajo de investigación presentado como requisito parcial para optar al título de:

Magister en Seguridad Informática

Director (a):

MSc. Alicia Osorio Builes

Director (a):

PhD. Paula Andrea Rodríguez Marín

Línea de Investigación:

Ciencias computacionales

Instituto Tecnológico Metropolitano

Facultad de Ingeniería

Medellín, Colombia

2021

*La receta para el **éxito**:*

***estudia** mientras los demás están durmiendo,*

***trabaja** mientras los demás están holgazaneando,*

***prepárate** mientras los demás están jugando,*

*y **sueña** mientras los demás están deseando*

William A. Ward

Agradecimientos

Agradezco a mis directoras de esta investigación PhD. Paula Andrea Rodríguez Marín y MSc. Alicia Osorio Builes, y también a Héctor Fernando Vargas Montoya, por el apoyo que me brindaron durante mi proyecto de grado. Así mismo, hago un reconocimiento a mis compañeros de estudio y profesores por sus aportes, a todos aquellos que aportaron su granito de arena para la construcción de este documento y a los que revisaron y aprobaron mi trabajo. Todo esto fue posible gracias a ustedes, por confiar en mí.

Resumen

La mayoría de las aplicaciones desarrolladas a la medida disponen de su propio mecanismo de autorización, lo que genera problemas para la gestión de los accesos de los usuarios. Tales como la asignación o retiro de los permisos en todas las aplicaciones oportunamente, la falta de auditoría y monitoreo en la gestión de los permisos y la dificultad para identificar los accesos que tiene un usuario, entre otros. Esta dificultad en la gestión de los accesos posibilita exponencialmente la consulta de información no autorizada. Algunas veces estas falencias mencionadas ocurren porque durante la implementación de una aplicación, no se especifican ni se desarrollan los requisitos de autorización facilitando al atacante a explorar vulnerabilidades que afectan la confidencialidad de la información e incluso la organización puede llegar a incumplir la Ley 1581 de 2012 que corresponde a la protección de datos personales. De acuerdo con lo descrito anteriormente, este proyecto de grado definió un modelo de autorización centralizado para ayudar a disminuir los problemas mencionados y realmente otorgarle beneficios a la organización.

Para lograrlo, este proyecto se dividió en cuatro fases: 1) Se conoció como estaba implementado el control de acceso de las aplicaciones desarrolladas a la medida en algunas organizaciones; 2) se identificaron algunos modelos de seguridad para el control de acceso más reconocidos en la industria, buscando en bases de datos científicas, en empresas como Gartner, NIST, ACIS, IEEE, y en proveedores de tecnología, entre otros; 3) se definieron los requisitos y el diseño del modelo centralizado de autorización en las aplicaciones a la medida; 4) se realizó un desarrollo donde se implementó parte del modelo centralizado de autorización, posteriormente se presentó a varias personas de TI con el fin de validar si ayudaba a reducir los problemas mencionados previamente y mitigar el riesgo de consultar información por personas no autorizadas.

Palabras clave: ABAC, Acceso, Autorización, Datos, Protección, RBAC, Seguridad en aplicaciones.

Abstract

Most of the applications developed to measure have their own authorization mechanism, which generates problems for the management of user access. Such as the assignment or withdrawal of permissions in all applications in a timely manner, the lack of auditing and monitoring in the management of permissions and the difficulty in identifying the accesses that a user has, among others. This difficulty in access management exponentially makes it possible to consult unauthorized information. Sometimes these shortcomings occur because during the implementation of an application, authorization requirements are not specified or developed, making it easier for the attacker to explore vulnerabilities that affect the confidentiality of the information and even the organization may breach Law 1581 of 2012 which corresponds to the protection of personal data. In accordance with the above, this degree project defined a centralized authorization model to help reduce the aforementioned problems and really provide benefits to the organization.

To achieve this, this project was divided into four phases: 1) It became known how access control was implemented for custom-developed applications in some organizations; 2) some of the most recognized security models for access control in the industry were identified, searching scientific databases, companies such as Gartner, NIST, ACIS, IEEE, and technology providers, among others; 3) the requirements and the design of the centralized authorization model in custom applications were defined; 4) A development was carried out where part of the centralized authorization model was implemented, subsequently it was presented to several IT people in order to validate if it helped reduce the problems raised and mitigate the risk of consulting information by unauthorized persons.

Keywords: ABAC, Access, Application security, Authorization, Data, Protection, RBAC.

Contenido

Lista de figuras

Pág.

Lista de tablas

Pág.

Lista de Símbolos y abreviaturas

Abreviatura	Término
ABAC	Attribute Based Access Control
CA	Criterios de aceptación
CapBAC	Capability Based Access Control
CBAC	Context Based Access Control
CWE	Common Weakness Enumeration
DAC	Discretionary Access Control
EY	Ernst & Young
HU	Historia de usuario
IAM	Identity and Access Management
ICONTEC	Instituto Colombiano de Normas Técnicas y Certificación
IEEE	Institute of Electrical and Electronics Engineers
JSON	JavaScript Object Notation
JWT	JSON Web Token
LDAP	Lightweight Directory Access Protocol
NIST	National Institute of Standards and Technology
OAUTH	Open Authorization
SOC	Security Operation Center
OWASP	Open Web Application Security Project
PAP	Policy Administration Point
PBAC	Path Based Access Control
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
RBAC	Role Based Access Control
SAML	Security Assertion Markup Language
SSO	Single Sign-On
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language

Introducción

Es claro que la información de las empresas que se tenía de forma física (papel) se está trasladando a formatos digitales, convirtiendo a las aplicaciones y sistemas de información en pilares para soportar estos volúmenes de datos. Esta es una de las razones por las que la seguridad de la información ha tomado fuerza para evitar accesos no autorizados, fuga de información, e indisponibilidad de los servicios, entre otros. Por estos riesgos de seguridad, es importante que los dueños de la información definan claramente las reglas para el control de acceso a la información y los sistemas deben acogerse a ellas. Así mismo, deben definir una periodicidad para la revisión de la autorización en las aplicaciones con el fin de que la información realmente sea accedida solo por las personas autorizadas y de esta manera se mitiguen los riegos de seguridad mencionados anteriormente [1].

Al desarrollar aplicaciones, los analistas de requisitos se enfocan generalmente en la parte funcional, es decir, los requisitos del usuario, y menos en los requisitos no funcionales como es el de la seguridad. Suelen no hacerse preguntas como ¿cuáles son los riesgos de exponer la información? ¿se utilizaron las últimas versiones? ¿se conoce al menos el top ten de vulnerabilidades en las aplicaciones? ¿se realizaron pruebas de seguridad a las aplicaciones antes de salir a producción? ¿se tiene conocimiento de las leyes y regulaciones que se debe tener presente en el desarrollo, por ejemplo, la Ley 1581 de 2012 sobre protección de datos personales? ¿se desarrolló la aplicación con el principio del mínimo privilegio con el fin de evitar que el impacto en la vulnerabilidad del sistema sea mínimo? Entre otras preguntas de seguridad que son importantes durante el ciclo de vida de desarrollo del software [2].

Cuando una empresa tiene varias aplicaciones desarrolladas a la medida y su proceso de autorización es de manera descentralizada, aumenta la administración y los posibles riesgos de seguridad asociados al desaprovisionamiento, genera problemas para el seguimiento en la gestión del control de acceso, aumenta exponencialmente los documentos no enlazados con los permisos asignados e incrementa las malas prácticas en la implementación del control de acceso, entre otros [3]. Los problemas acá mencionados se encuentran definidos como controles que debe tener un sistema de información según la ISO 27002 en el dominio de control de acceso [4]. Estos factores permiten a un atacante llevar a cabo acciones fraudulentas, acceder a información que no está

autorizado o simplemente causar daños reputacionales, financieros o incluso sociales a las empresas [5].

Según el reporte bimensual de Contrast Security para los meses de enero y febrero del 2020, las vulnerabilidades más frecuentes se encuentran en las aplicaciones desarrolladas al interior de la organización. El reporte muestra que el 47% de las aplicaciones que monitoreó durante enero-febrero de 2020 tuvieron vulnerabilidades críticas. El 31% de las aplicaciones Java presentaron la vulnerabilidad de XSS (Cross Site Scripting) y el 12% de las aplicaciones tuvieron en promedio la vulnerabilidad de Sql Injection [6]. Así mismo, en el informe de ElevenPaths en el año 2019 reportó 408 incidencias relacionadas con el control de acceso incorrecto (CWE 284) y 238 incidencias con mal manejo de permisos, privilegios y control de acceso (CWE 264)[7]. En otro informe de la empresa Positive Technologies, indicó que en el año 2019, el acceso no autorizado a las aplicaciones es posible en el 39% de los sitios y que el 82% de las vulnerabilidades se ubicaron en el código de la aplicación [8]. Estas estadísticas motivaron al diseño del modelo centralizado de la autorización en las aplicaciones con el propósito de ayudar a mitigar los riesgos de seguridad a las empresas frente al manejo de información .

Este trabajo se enmarca en el siguiente objetivo general:
Diseñar un modelo centralizado de autorizaciones para mejorar la seguridad en aplicaciones desarrolladas a la medida en las organizaciones con el fin de disminuir malas prácticas de desarrollo en el control de acceso, el robo de información, el desaprovisionamiento inoportuno y la falta de seguimiento y monitoreo.

Para lograr este objetivo, se plantearon los siguientes objetivos específicos:

- Identificar cómo manejan la autorización en las aplicaciones a la medida y su gobierno, en al menos cuatro organizaciones.
- Determinar cuáles son las mejores prácticas para el manejo de la autorización en las organizaciones.
- Establecer los requerimientos mínimos requeridos para el control de acceso en las aplicaciones a la medida, en cuanto a la autorización.
- Validar el modelo de seguridad a través de una prueba de concepto, que permita establecer el nivel de cumplimiento en el control de acceso.

El alcance de este proyecto de grado es diseñar un modelo centralizado del control de acceso para mejorar la seguridad en las aplicaciones desarrolladas a la medida con el fin de disminuir las vulnerabilidades durante la ejecución o uso de las aplicaciones y que pueda ser utilizado por cualquier tipo de empresa pública, privada, mixta, o incluso de cualquier sector empresarial. Empresas como Microsoft, Oracle, IBM entre otros grandes líderes ofrecen un IAM Identity Access Management para facilitar la autenticación y la autorización en las aplicaciones empresariales sin embargo, estas pueden ser costosas o difíciles de implementar. [9], [10].

Frente al tema de limitaciones del proyecto se parte del hecho que el tema de autenticación ya está resuelto en las organizaciones y este proyecto de grado solo definirá el modelo para la autorización en las aplicaciones desarrolladas a la medida. Así mismo, para la validación del modelo centralizado se desarrolló una aplicación con las características mínimas requeridas para proceder a la evaluación de este por algunas de las empresas entrevistadas, dicha aplicación no es del todo funcional ni fue implementado con las prácticas de seguridad, en caso de utilizarse este modelo se debe desarrollar desde cero.

Este trabajo se divide en los siguientes capítulos:

- Marco teórico y estado del arte.
- Metodología utilizada para cumplir con los objetivos específicos.
- Los resultados por cada uno de los objetivos específicos.
- Las conclusiones y recomendaciones.
- Los anexos o documentos de soporte del proyecto.

1. Marco Teórico y Estado del Arte

1.1. Marco teórico

El presente trabajo pretende diseñar un modelo centralizado de autorizaciones para mejorar la seguridad en aplicaciones desarrolladas a la medida en las organizaciones que les permita gestionar el mecanismo de autorización con el fin de proteger su información. Por lo tanto, se hace necesario aclarar algunos conceptos dentro de este documento.

La seguridad de la información se compone de tres grandes pilares que son: Disponibilidad, Integridad y Confidencialidad, los cuales son definidas en la norma ISO 27000 de 2013: Disponibilidad, se refiere a la propiedad de ser accesible y utilizable a demanda por una entidad autorizada; Integridad, se define como la propiedad de exactitud y completitud; y por último, Confidencialidad, que se entiende como la propiedad de la información que la hace no disponible o sea divulgada a individuos, entidades o procesos no autorizados [11].

Un modelo se define como una forma de describir un conjunto dado de componentes y cómo esos componentes se relacionan entre sí para describir el funcionamiento principal de un sistema [12]. Ahora, ¿Qué se entiende por Autorización?, es el proceso que garantiza que los usuarios correctamente autenticados puedan acceder solo a aquellos recursos para los cuales el propietario les ha dado su aprobación [13]. La autorización también se conoce como control de acceso, y éstas dos palabras serán usadas de aquí en adelante como sinónimos dentro de este documento. El control de acceso puede ser utilizado en diferentes contextos como: proteger los archivos y carpetas en un equipo de cómputo, controlar el acceso a las bases de datos, preservar la información en las aplicaciones, este último es el alcance para este documento [14].

Así mismo, la seguridad de la información en las organizaciones se logra en la medida en que se definan controles, políticas, procedimientos y procesos. Según las definiciones hechas por el ICONTEC en la ISO 27000, Control se define como la medida que modifica un riesgo; Política se refiere a intenciones y dirección de una organización, como las expresa formalmente su alta dirección; Procedimiento es el método de ejecución, y, por último, sin ser el menos importante, Proceso se relaciona con el conjunto de actividades interrelacionadas o que interactúan, que

transforma entradas en salidas. Para poder medir la efectividad y cumplimiento de los controles definidos en un sistema de gestión de la seguridad de la información, se hace necesario realizar auditorías periódicas, entendiendo auditoría como un proceso sistemático, independiente y documentado para obtener evidencias y evaluarlas de manera objetiva con el fin de proteger sus activos y recursos [11]. La evaluación permite la mejora continua en los procesos y procedimientos definidos en la organización y la prestación de un mejor servicio a los clientes.

Otro aspecto importante es la Gestión de Identidad y Acceso (IAM) el cual tiene como objetivo proporcionar un único mecanismo de autenticación y control de acceso a las aplicaciones en las organizaciones. Si cada aplicación tiene el mismo usuario y contraseña, se puede facilitar el inicio de sesión único (SSO) entre ellas y utilizar un modelo de autorización centralizado, tales como: Control de acceso discrecional (DAC), el Control de acceso basado en roles (RBAC) y el Control de acceso basado en atributos (ABAC) [15], [16], [17], [18], [19], para mantener la administración de los permisos de los usuarios en los sistemas de manera oportuna y en tiempo real.

En las organizaciones, las aplicaciones desarrolladas a la medida son soluciones que satisfacen las necesidades específicas o requerimientos de un grupo de usuarios. Los requerimientos en las metodologías tradicionales como CMMI, RUP el término requerimiento es muy utilizado, pero en metodologías ágiles se conocen como necesidades o historias de usuario la cual tienen la misma finalidad que es determinar qué hará el software. Estas aplicaciones son desarrolladas en lenguajes de programación como C, Java y .Net, el cual la organización es dueña del código fuente y que requieren del uso de un IAM para lograr una única identidad del usuario y su correcta gestión del control de acceso. Se hace una diferenciación entre dos tipos de aplicaciones que se utilizaran dentro de este documento que son aplicación de negocio, la cual se entiende como aquellas aplicaciones que hacen uso del componente centralizado de autorización donde tienen definido el control de acceso y la aplicación de control de acceso centralizado que corresponde al componente encargado de administrar la autorización para las aplicaciones de negocio o desarrolladas a la medida.

1.1.1. Modelos de control de acceso

En el año 1941, el ingeniero civil Konrad Zuse inventó la primera computadora donde podía guardar hasta 64 palabras, su objetivo era automatizar los cálculos estadísticos [20]. Desde allí empezó una nueva era para los sistemas de información, en los años sesenta, se crearon las primeras aplicaciones con el fin de optimizar el tiempo en los cálculos de los procesos de contabilidad, facturación y nómina en las organizaciones. Con el paso del tiempo, se incorporaron más procesos, sin embargo, se generó tanta información aislada que fue difícil procesarla y los empresarios se preguntaban por qué teniendo los datos en un computador, no todos podían acceder a ella, y es finalizando los años setenta donde se crearon las primeras bases de datos con el fin de tener información centralizada y coherente y así adquirir una ventaja competitiva de manera eficaz y eficiente en la organización [21]. Posteriormente, se comenzó a hablar sobre los problemas de seguridad de los datos. Los administradores de los sistemas tenían el reto de garantizar que las personas sólo pudieran ver la información a la que estaban autorizadas, para ello definieron varios tipos de control de acceso y uno de los que surgió fue el modelo *Role Based Access Control* RBAC, donde los administradores crearon roles de acuerdo con las funciones del empleado con el fin de permitir o rechazar el acceso a la información [22]. Con el paso del tiempo, surgió otro modelo de control de acceso llamado Control de Acceso Basado en Atributos, por sus siglas en inglés ABAC [19], y así sucesivamente con la evolución de la tecnología. Estos modelos y otros están detallados en el capítulo 3.2.1.

1.1.2. Metodologías ágiles

Las metodologías ágiles son marcos de trabajo basados en la confianza y la colaboración entre el cliente, tecnologías de información y el proveedor, orientadas a lograr entrega temprana de valor a los clientes. Una de las metodologías es SCRUM, el cual consta de procesos iterativos e incrementales y enfocado en mejorar continuamente el producto y el equipo de trabajo. Todo el equipo SCRUM trabaja en conjunto para entregar de manera temprana productos funcionando [23], [24].

A continuación se describen algunos términos utilizados en este documento [24]

- ✓ Historia de Usuario: corresponde a las necesidades, funcionalidades o requisitos del producto, de acá en adelante HU.
- ✓ Backlog de producto que corresponde al listado de necesidades del cliente para cumplir con la visión del producto.
- ✓ Product Owner: es el dueño, doliente del problema y responsable de la priorización de las historias de usuario.

Algunos de los principios ágiles son "Satisfacer a los clientes con entregas tempranas y continuas de software con valor", "Adaptarse a los cambios en cualquier momento", "El software funcionando es la principal medida de progreso" [25]. Mientras que las metodologías tradicionales se van quedando atrás porque no hay retroalimentación al equipo, no existe mejora continua, no se tiene constante comunicación con el cliente, puede ocurrir que lo que necesita el cliente al final no es lo que se implementa. Es por lo anterior, que para este proyecto se utilizó la metodología SCRUM porque permite de manera temprana entregar productos funcionando e ir mejorándolo y evolucionándolo [5], [26].

1.1.3. Estadísticas de vulnerabilidad en autorización de aplicaciones

El CWE - Common Weakness Enumeration es una lista creada por el Mitre con el fin de enumerar las debilidades o vulnerabilidades que se pueden encontrar en un software o hardware. Está información va dirigida a personal que trabaje en el área de sistemas o seguridad [27]. Dentro del listado de vulnerabilidades reportadas se encuentran las siguientes que son factores claves en este proyecto:

- ➢ CWE-264: Permisos, privilegios y controles de acceso. Las debilidades en esta categoría están relacionadas con la administración de permisos, privilegios y otras características de seguridad que se utilizan para realizar el control de acceso [28].

- CWE-284: Control de acceso incorrecto. El software no restringe o restringe incorrectamente el acceso a un recurso de un usuario no autorizado. En la Tabla 1-1 se relaciona algunas vulnerabilidades de alto nivel para esta categoría [29]:

Tabla 1-1 Vulnerabilidades relacionadas con el CWE-284.

CWE	Nombre
269	Gestión de privilegios inadecuada
282	Gestión de propiedad inadecuada
285	Autorización inapropiada
286	Gestión de usuario incorrecta
1220	Granularidad insuficiente del control de acceso

Fuente: [29]

- CWE-285: Autorización incorrecta. El software no realiza o realiza incorrectamente una verificación de autorización cuando un usuario intenta acceder a un recurso o realizar una acción. En la Tabla 1-2 se relaciona algunas debilidades de alto nivel de un software para esta categoría [30]:

Tabla 1-2: Vulnerabilidades relacionadas con el CWE-285.

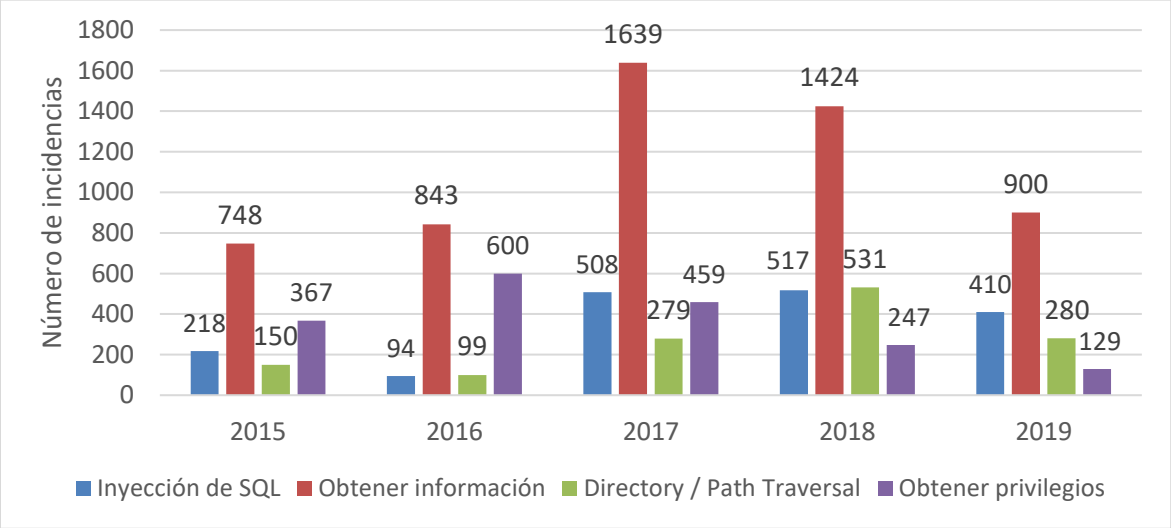
CWE	Nombre
552	Archivos o directorios accesibles a terceros
732	Asignación incorrecta de permisos para recursos críticos
862	Autorización faltante
863	Autorización incorrecta
1230	Exposición de información confidencial a través de metadatos
1244	Autorización inadecuada para la depuración física y las interfaces de prueba

Fuente: [30]

Estas vulnerabilidades pueden ser mitigadas desde las fases del ciclo de desarrollo de software: requisitos, arquitectura, diseño e implementación, definiendo el modelo de autorización a utilizar (RBAC, ABAC, entre otros).

Por otro lado, en el sitio web www.cvedetails.com se listan las vulnerabilidades clasificadas por proveedor, productos, fecha o tipo de vulnerabilidad. La fuente de la información es provista por la NIST - National Institute of Standards and Technology[1]. A continuación, en la Figura 1-1 se relacionan los tipos de vulnerabilidad por año que pueden resultar de la mala implementación de la autorización en las aplicaciones, allí se puede observar que la categoría "Obtener o ganar privilegios" es la que presentó mayor número de incidencias en los últimos 5 años, en segundo lugar continúa "Obtener privilegios" que le permite al atacante realizar operaciones con un nivel de privilegio superior al mínimo requerido y afectando la confidencialidad de la información:

Figura 1-1. Vulnerabilidades por tipo en los últimos cinco años.

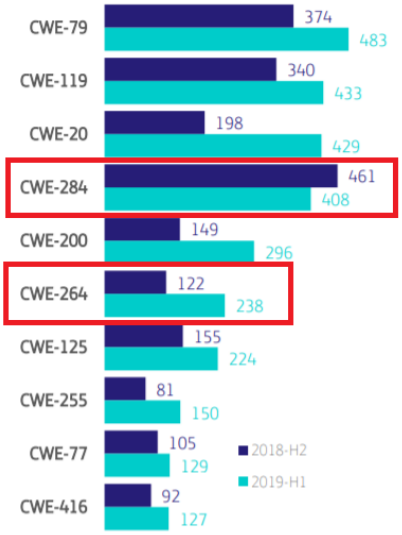


Fuente: [31]

En el informe sobre el estado de la seguridad del año 2019 elaborado por ElevenPaths se muestran los CWE más frecuentes en los años 2018 y 2019, para más detalle ver Figura 1-2; allí se logra observar que el CWE-284 obtuvo un total de 408 incidencias y el CWE-264 obtuvo 238 incidentes en el año 2019. En la actualidad, siguen existiendo estás vulnerabilidades por descuido o desconocimiento del desarrollador [7].

[1] https://www.nist.gov/

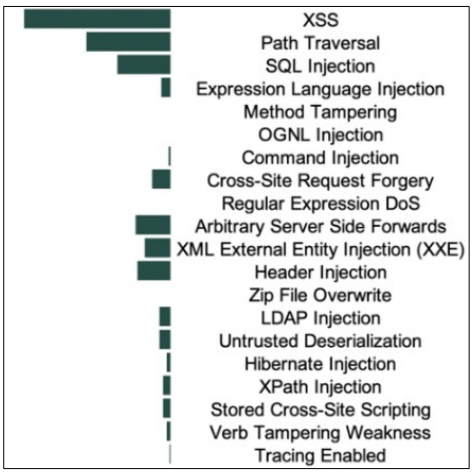
Figura 1-2 Top 10 CWE 2019.



Fuente: [7]

Otro informe realizado por la empresa Contrast Security[2] en diciembre de 2019 mostró las vulnerabilidades más comunes en las aplicaciones, donde las tres primeras son XSS, *Path Traversal* y SQL *Injection* [32], tal y como se puede ver en la Figura 1-3.

Figura 1-3 Probabilidad por vulnerabilidad 2019.

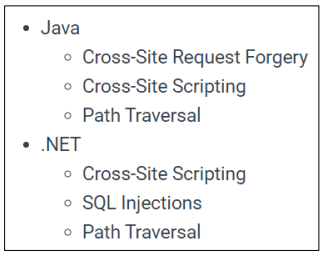


Fuente: [32]

[2] https://www.contrastsecurity.com/

En ese mismo reporte de Contrast Security, las 5 vulnerabilidades más frecuentes durante el desarrollo de software son: XSS, *Path Traversal*, *SQL Injection*, *Arbitrary Server Side Forwards y Header Injection*. También presentó las vulnerabilidades según el lenguaje de programación: Java o .Net [32], eso se puede apreciar en la Figura 1-4:

Figura 1-4 Vulnerabilidades por lenguaje de programación Java y .Net.



Fuente: [32]

En la actualidad, las organizaciones están invirtiendo dinero en la adquisición de herramientas o servicios que apoyen a la ciberseguridad y puedan proteger su activo más importante que es la información. Es así como en la encuesta global de seguridad de la información realizada por el proveedor especializado en seguridad Ernst & Young Global Limited [33], muestra que las compañías más grandes tienen más probabilidad de aumentar sus presupuestos en el año 2018 (63%) y el 2019 en un 67%, mientras que las pequeñas organizaciones lo harían en 50% y 66% respectivamente. En la Figura 1-5, se logra observar en porcentajes del presupuesto como las empresas invertirán en temas de ciberseguridad para este y el próximo año.

Figura 1-5 Presupuesto estimado de las empresas.

¿Cómo cambió el presupuesto total de ciberseguridad de las organizaciones este año? ¿Cómo cambiará para los próximos 12 meses?

	Este año	El próximo año
Incrementará más del 25%	12%	15%
Incrementará más del 15 y 25%	16%	22%
Incrementará más del 5% y 15%	25%	28%
Se mantendrá prácticamente igual (entre +5% y -5%)	40%	31%
Disminuirá entre 5% y 15%	4%	2%
Disminuirá entre 15% y 25%	1%	1%
Disminuyó entre 25%	1%	1%

Fuente: [33]

1.2. Estado del arte

Para este proyecto de grado se consultaron las siguientes bases de datos científicas: Arxiv, Dialnet, Forrester, Gartner, IEEE, Nist, SANS y Thompson Reuters. Así mismo, comunidades o proyectos de seguridad como el Mitre y OWASP, y algunas empresas reconocidas en la industria, por ejemplo: Contrast Security, ElevenPaths, EY, IBM, Icontec, Microsoft y Oracle. La investigación se realizó durante el período de años comprendido entre el 2013 al 2020. Adicional, para hacer esta búsqueda se utilizaron las palabras claves: control de acceso, autorización, gestión de acceso de identidad, perfiles, roles, control de acceso en Blockchain, estándares de autorización para desarrollo de aplicaciones. En la Tabla 1-3 se indicaron los criterios de inclusión y exclusión para los artículos encontrados.

Tabla 1-3. Criterios de inclusión y exclusión de artículos o proyectos.

Criterios de inclusión	Criterios de exclusión
Aplicaciones web desarrolladas a la medida	Artículos inferiores al año 2013
Artículos o proyectos escritos sólo en inglés o español.	Artículos diferentes a los idiomas inglés o español.
Modelos de control de acceso para aplicaciones web.	Controles de acceso a recursos como servidores o bases de datos o incluso en la nube.
Prácticas de desarrollo seguro para controles de acceso	Temas relacionados con sicología, abogacía, historia, geografía.
Vulnerabilidades actuales en las aplicaciones	Artículos relacionados solo con temas de autenticación.
Temas relacionados con Tecnologías de Información	Temas relacionados con Tecnologías de operación como sistemas SCADA.

Con base en las búsquedas realizadas, se detallaron a continuación los artículos o documentos más relevantes en el estado del arte.

La transformación digital es un factor clave en las organizaciones hoy en día y, para lograrla, se debe apoyar en tecnología, procesos y personas. Un habilitador para esto es la gestión de acceso

de identidad, conocido por sus siglas en inglés IAM - Identity Access Management, el cual se encarga del aprovisionamiento, desaprovisionamiento, autenticación, autorización, inicio de sesión único y monitoreo de las cuentas de los usuarios [34], [35]. En el mercado se encuentran varios proveedores que ofrecen soluciones que apoyan la centralización de la autorización de las aplicaciones tales como SailPoint, Oracle, IBM, One Identity y Saviynt; estas empresas se encuentran como líderes en el cuadrante mágico de Gartner [9], y pueden ser la base para resolver el problema planteado porque buscan que exista una sola identidad y control de acceso del usuario en las organizaciones [36], sin embargo, se deben evaluar frente a aspectos como necesidades o requerimientos del modelo a definir en este proyecto de grado, su costo e implementación, requerimientos de negocio, facilidad de uso y su mantenimiento, que es allí donde las soluciones se quedan cortas porque no permite personalizaciones y las empresas deben adoptarse a lo existente.

El Instituto Nacional de Estándares y Tecnología, por sus siglas en inglés NIST, es una empresa de Estados Unidos con el propósito de definir guías, frameworks, normas, mejores prácticas, controles entre otros basados en el uso de la Tecnología. Algunos artículos que fueron importantes para este proyecto:

- NIST SP 800-53 rev 4: Security and Privacy Controls for Federal Information Systems and Organizations [37].
- NIST 800-162: *Guide to Attribute Based Access Control (ABAC) Definition and Considerations* [17].
- NIST 800-205: *Attribute Considerations for Access Control Systems* [18].

Algunos elementos mencionados en las referencias anteriores ayudaron en el diseño del modelo centralizado de autorización para proteger la información de forma que sea accesible solo por las personas adecuadas; sin embargo, se debió complementar con otros modelos para que abarcara mayor contexto y seguridad.

El Proyecto Abierto de Seguridad en Aplicaciones Web (OWASP) es una comunidad abierta dedicada a permitir que las organizaciones desarrollen, adquieran y mantengan sus aplicaciones seguras. En el año 2017 publicó los diez riesgos más críticos en las aplicaciones web [38], dentro de

los que se encuentra el riesgo A5 - Pérdida de control de acceso, el cual se relaciona con la mala implementación del control de acceso. Hoy en día, las aplicaciones siguen siendo vulnerables frente a este control por no tener prácticas de desarrollo seguras definidas en la organización [16].

Hoy en día aún se leen noticias donde se encuentra que fueron robadas las claves de los usuarios y esto genera preocupaciones frente al tema de la confidencialidad de la información. Por ejemplo, el caso de Facebook sobre la publicación de información no autorizada por el usuario, es un caso de una incorrecta implementación en el control de acceso, donde Facebook solicitaba el número de celular para poder activar el doble factor de autenticación, el usuario seleccionaba que ese campo solo fuese visible por el dueño y, sin embargo, los servicios web que exponía Facebook permitían que ese campo fuera visible para otras personas, incluso que Google o Pipl lo indexara en sus bases de datos y cualquier persona tuviera acceso a éste [39]. Asimismo, Gartner también cree que a raíz de lo anterior, las empresas impulsarán al menos un 10% de la demanda frente a temas de ciberseguridad, donde algunos componentes impactados son: Gestión de acceso de identidad (por sus siglas en inglés IAM), gestión y administración de identidad (por sus siglas en inglés IGA) y la prevención de pérdida de datos (por sus siglas en inglés DLP) [40].

La NIST publicó en el año 2019 el artículo [18], el cual pretende entregar unas consideraciones en el uso del modelo de control de acceso basado en atributos desde el punto de vista del diseño y su implementación, también menciona que los atributos deben establecerse, emitirse, almacenarse y administrarse bajo una autoridad o gobernanza. Aunque el documento publicado por la NIST es bueno solo aplica para el modelo de autorización ABAC por lo que para este proyecto de grado no es suficiente tener uno solo.

Según el artículo [41] publicado en la IEEE, los autores buscan resolver un problema crítico de control de acceso en Internet de las cosas, en adelante IoT, donde pretenden definir un marco inteligente basado en contratos, el cual consiste en múltiples contratos de control de acceso (ACC), un contrato de juez (JC) y un contrato de registro (RC), para lograr un control de acceso distribuido y confiable para los sistemas IoT. Sin embargo, este artículo no resuelve el problema planteado en este proyecto porque está enfocado a aplicaciones desarrolladas a la medida en las organizaciones.

En el artículo [42] del año 2016, los autores Indu y Rubesh Anand diseñaron un modelo IAM híbrido para aplicaciones web conformado por los modelos de control de acceso DAC, RBAC, ABAC apoyados de la tecnología SAML2.0, los grupos de usuarios son creados en el directorio activo y la solicitud de acceso al grupo se realiza a través del Sistema de solicitud de acceso (ARS), su enfoque es interesante y puede resolver el problema planteado por la combinación de varios modelos de control de acceso, no obstante, el directorio activo no permite definir roles y puede llegar a tener limitantes en el tamaño del token cuando se entregue a la aplicación porque un usuario podría llegar a estar en muchos grupos; adicional, su administración puede ser compleja y difícil de identificar esos grupos a que aplicaciones pertenece y con qué rol; otra limitante es la tecnología SAML debido a que esta se usa principalmente al interior de las organizaciones pero para conectarse con aplicaciones publicadas en internet se utiliza OAUTH.

Por otro lado, los autores Kononov y Isaev, en el artículo [15] buscan el desarrollo de un modelo de control de acceso de seguridad para aplicaciones web basado en rutas, que amplía el modelo RBAC original y permite un control de acceso flexible mediante la ruta de solicitud web. Aunque tiene una gran aproximación a la problemática definida en esta propuesta de grado, no resuelve el problema del modelo centralización de autorización en las aplicaciones desarrolladas a la medida y además, la propuesta no apoya la protección de datos personales porque se basa en la URL.

El texto [43] del año 2017 ilustra un enfoque de interoperabilidad para intercambiar información entre la aplicación desarrollada con el framework Spring y utilizando SecureUML, modelos de sistemas distribuidos seguros basados en el lenguaje de modelado unificado UML, para definir el modelo RBAC. Aunque utiliza buenas prácticas de desarrollo seguro, este modelo solo está definido para controles de acceso basado en roles y no se define como un modelo centralizado sino que cada aplicación desarrolle su propio mecanismo.

Con respecto al estándar XACML [44] es un lenguaje basado en XML con el propósito de declarar las políticas de seguridad y control de acceso en las aplicaciones, su uso principal es para modelos tipo ABAC, es un buen candidato para el modelo centralizado que se pretende definir en este proyecto pero como desventaja tiene que, por defecto, no está integrado con un modelo RBAC y su formato es XML lo que dificulta su lectura y escritura en la aplicación, su estructura es estricta y

no permite error en su código frente a un formato tipo JSON el cual está formado por cadena: valor, es más liviano y su tamaño es menor.

Otro artículo relacionado con el tema de investigación fue [45] el cual propone un modelo de control de acceso dinámico, es decir, tomar decisiones en tiempo real y la aplicación de políticas por medio de flujos de trabajo para garantizar la gobernabilidad y la responsabilidad de la autorización en la aplicación, aunque mejora el modelo RBAC se requiere una aprobación previa para poder otorgarse el acceso. Sin embargo, no soluciona la problemática planteada en este documento, que pretende definir un modelo centralizado de control de acceso para aplicaciones desarrolladas a la medida y el enfoque planteado solo cubre el sector financiero.

De los artículos mencionados anteriormente, se seleccionaron los 5 más relevantes, se indica un resumen del artículo, cuáles son sus limitaciones y los aportes para planteamiento del problema de este proyecto de grado, éstos se pueden observar en la Tabla 1-4. Estos documentos fueron clave en la parte de la definición del modelo en el objetivo 3, de igual forma, se puede observar que no existe un único documento que resuelva el planteamiento del problema de este proyecto de grado, no obstante, la suma de ellos puede aportar para construir el modelo ideal.

Tabla 1-4. Cuadro comparativo del estado del arte.

Trabajo relacionado	Propuesta	Limitantes	Aportes
2019: NIST SP 800-205, Attribute Considerations for Access Control Systems[18].	Proporciona una guía para la definición de la autorización por medio del mecanismo ABAC en las aplicaciones.	Está basado en un solo modelo.	Sirve de referencia para uno de los modelos de control de acceso para este proyecto de grado.
2018: Improving Web Applications Security Using Path-Based Role Access Control Model [15].	Modelo de control de acceso de seguridad para aplicaciones web basado en rutas, que amplía el modelo RBAC original y permite un control de	No resuelve el problema del modelo centralización de autorización en las aplicaciones desarrolladas a la medida	Este nuevo modelo puede aportar para definir el modelo centralizado de autorización.

Trabajo relacionado	Propuesta	Limitantes	Aportes
	acceso flexible mediante la ruta de solicitud web.	y la propuesta no apoya a la protección de datos personales porque se basa en la URL.	
2017: An Approach to Capture Role-Based Access Control Models from Spring Web Applications [43].	En este artículo se propone un marco de interoperabilidad para analizar el código de las aplicaciones web desarrolladas con el framework y seguridad de Spring basado en el modelo de control de acceso RBAC.	Está enfocado solo en el modelo de control de acceso basado en roles y no está diseñado para ser centralizado.	Puede aportar sobre el mecanismo de desarrollo para validar el control de acceso basado en RBAC y complementar el modelo centralizado del proyecto.
2016: Hybrid Authentication and Authorization Model for Web based Applications [42].	Define un modelo híbrido propuesto que ayuda a las organizaciones a implementar la identidad, las políticas de gobierno y el control dinámico de acceso de los usuarios.	El modelo abarca varios componentes de un IAM y la propuesta de este proyecto es solo autorización	El modelo definido en este artículo puede aportar para la propuesta final de este proyecto de grado.
2013: eXtensible Access Control Markup Language (XACML) Version 3.0 [44].	Especificación técnica del modelo XACML basado en atributos del usuario.	El modelo devuelve la información en formato XML y este puede ser complejo frente a otros formatos como JSON	El modelo definido en este artículo puede aportar para la propuesta final de este proyecto de grado.

Fuente: Propia.

Algunos factores diferenciadores frente a herramientas existentes en el mercado son:

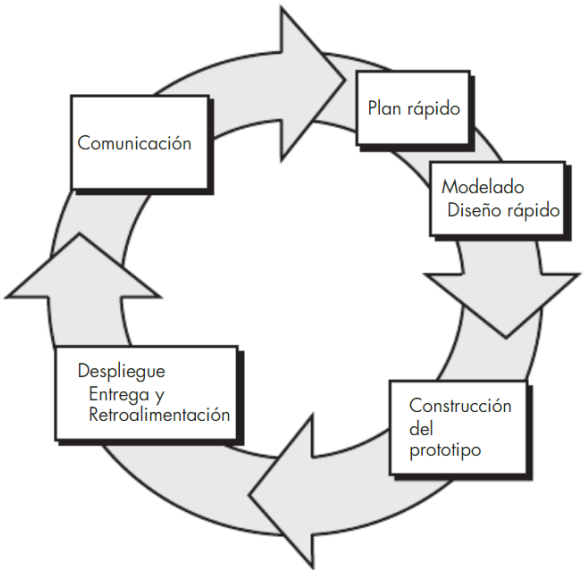
1) El costo, debido a que empresas como Oracle, IBM, Microsoft que son líderes en los cuadrantes mágicos de Gartner, ofrecen soluciones de alto costo que no son fácilmente alcanzables para ciertas empresas.
2) Su implementación, por la simplicidad de este diseño no se requiere tanta infraestructura ni inversiones en tiempo adicional para aprender las herramientas que se adquieran.
3) Se ajusta a la necesidad de la empresa al desarrollar aplicaciones a la medida acorde a los requisitos específicos de la Organización.
4) Modelo centralizado para aplicaciones desarrolladas, los paquetes comprados tienen resuelto el tema de autorizaciones, sin embargo, en términos generales, cuando se desarrolla a la medida alguna necesidad, de entrada, no se contempla la implementación de los temas relacionados con los accesos.

De acuerdo con lo mencionado en el estado del arte, se puede concluir que a pesar de existir documentación similar como artículos, tesis, informes, entre otros, al problema planteado en este proyecto de grado, ninguna de ellas está enfocada a aplicaciones desarrolladas a la medida ni tampoco a tener centralizado para todas estas aplicaciones el control de acceso. Así mismo, se encontró que, aunque existen varios mecanismos de autenticación y autorización definidos, las organizaciones, por desconocimiento, por costo o por no tener buenas prácticas de desarrollo seguro, no implementan los controles de acceso y por eso hoy en día, se continúa mostrando vulnerabilidades frente a este tema como se observa en el Top Ten de OWASP. El objetivo principal de esta propuesta es definir un modelo centralizado de autorizaciones para mejorar la seguridad con algunas herramientas que apoyen su implementación para que las empresas puedan utilizarlo como referencia en las aplicaciones desarrolladas a la medida.

2. Metodología

Para cumplir el objetivo general de este proyecto de grado se definió una metodología basada en prototipos porque permite de manera evolutiva ir adaptando el modelo con base en las nuevas sugerencias sobre lo presentado hasta llegar al definitivo y así mismo ayudar a mejorar la comprensión de la problemática a resolver. El proceso comienza con una comunicación donde se reúnen los interesados para definir los requisitos de su necesidad, se realiza un plan rápido para hacer el prototipo, se define un diseño inicial para su posterior construcción y finalmente se realiza el despliegue, entrega y retroalimentación para mejorar la necesidad. Este ciclo se repitió tantas veces fueron necesarios hasta llegar al producto final de este trabajo de grado [46], este ciclo se puede encontrar descrito en la Figura 2-1. Cada una de las fases definidas en este proyecto de grado se trabaja por medio de prototipos, donde cada entregable se revisa y ajusta según las recomendaciones indicadas hasta llegar al producto final definido en cada de ellas.

Figura 2-1. Metodología basada en prototipos.

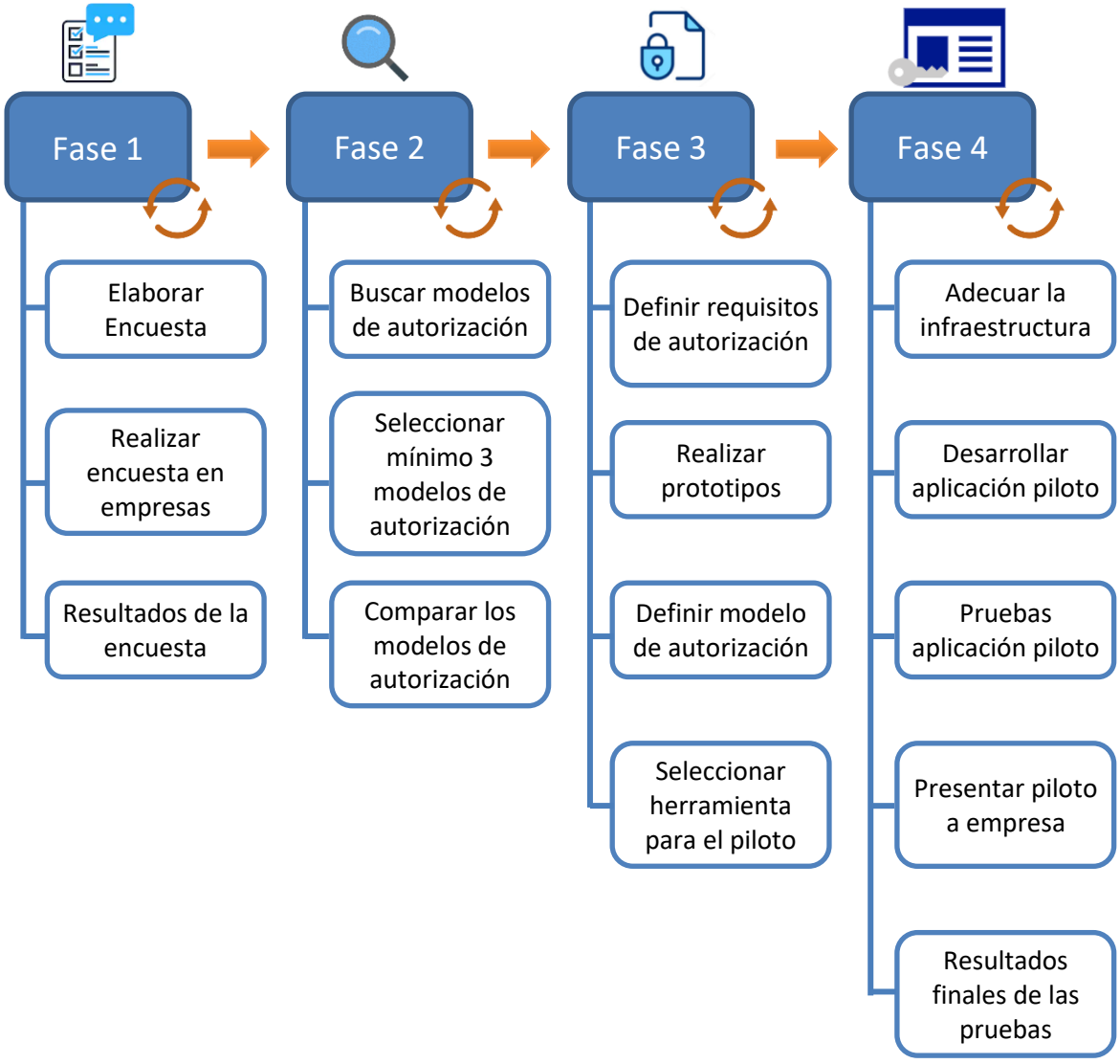


Fuente: [46]

En el plan rápido se bosquejaron 4 fases que equivalen a los 4 objetivos específicos, cada una a su vez tiene unas tareas y entregables que apoyan al cumplimiento del objetivo principal.

A continuación, se describe en resumen cada una de las tareas por cada fase, ver Figura 2-2.

Figura 2-2 Metodología de desarrollo del proyecto de grado.

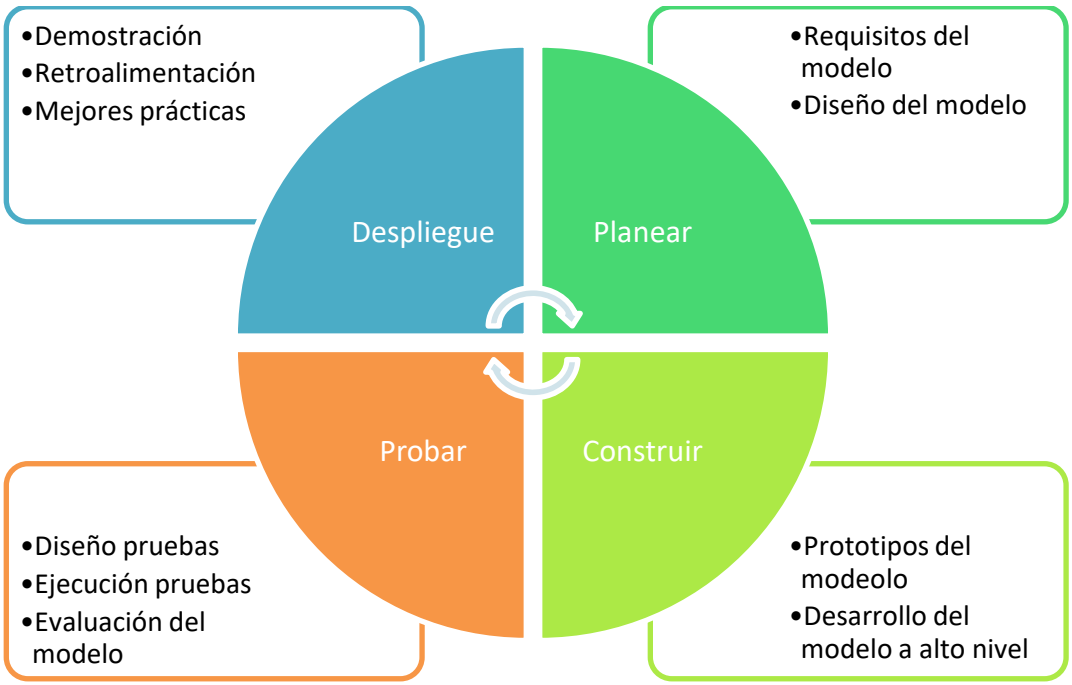


Fuente: Propia.

Para la validación y verificación del cumplimiento para el objetivo general se utilizó el diagrama mostrado en la Figura 2-3, el cual se dividió en cuatro etapas: Planear, Construir, Probar y Desplegar. Para cada una se definieron las tareas principales que se llevaron a cabo. Sigue siendo

un proceso continuo y finalizó cuando todos los interesados estuvieron de acuerdo con el modelo centralizado de autorización para aplicaciones desarrolladas a la medida.

Figura 2-3. Verificación objetivo general.



Fuente: Propia.

A continuación, se detalló por cada una de las fases u objetivos específicos el esquema de cumplimiento de estos.

2.1. Fase 1: Identificación de cómo manejan la autorización en las aplicaciones a la medida y su gobierno, en al menos cuatro organizaciones.

Esta primera fase tuvo como propósito diseñar la encuesta para identificar el estado actual de la autorización o control de acceso en las aplicaciones desarrolladas a la medida y poder reflejar si el planteamiento del problema existe o no en al menos cuatro (4) organizaciones, sin embargo, se logró hacer la encuesta en once (11) de ellas.

La encuesta fue diseñada con 14 preguntas, las cuales fueron enfocadas en los problemas que se pretenden resolver en este documento, divididas en 5 secciones: Gobierno, Arquitectura, Requisitos y diseño, Operación y Auditoría y seguimiento. Las preguntas fueron de selección múltiple con una o varias respuestas, un modelo del tipo de pregunta se puede observar en la Figura 2-4. Se realizaron varias revisiones de la encuesta hasta llegar a la definitiva. Para ver el detalle de la encuesta ir al Anexo A.

Figura 2-4. Ejemplo de una pregunta de la encuesta.

1. ¿La empresa tiene definida alguna política de control de acceso o autorización a nivel corporativo para las aplicaciones desarrolladas a la medida o para paquetes comprados? Seleccione la que más aplique.

☐	Se cuenta con un documento general de políticas de seguridad, pero no específicamente para aplicaciones.
☐	Se cuenta con un documento general de políticas de seguridad y se abordan allí las aplicaciones de forma general.
☐	Se cuenta con un documento general de políticas de seguridad, se abordan las aplicaciones hasta el nivel de control de acceso.
☐	Se cuenta con un documento general de políticas de seguridad, se abordan las aplicaciones y se cuenta con procedimiento claros y definidos de cómo asignar roles y perfiles, como retirar y cambiar, así como la revisión periódica de éste.
☐	Se tienen procedimientos de cómo asignar roles y perfiles, como retirar y cambiar, así como la revisión periódica de éste, pero no hay una política global sobre el tema.
☐	No se tiene un documento de políticas de seguridad.

Fuente: Propia.

Debido a que en la encuesta se recolectaba información sensible como datos personales y procesos de TI que usan las empresas, se incluyó un párrafo relacionado con la protección de los datos personales, por lo cual, dicha información no se mostrará como resultado en este proyecto.

Posteriormente, se identificaron posibles candidatos a ser entrevistados de diferentes organizaciones, se acordó con ellos un espacio para responder las preguntas, se envió previamente la encuesta para que pudieran analizarla, luego se llevaron a cabo las entrevistas, las personas diligenciaban la encuesta y la enviaban. Se consolidaron el total de encuestas para su posterior tabulación en Excel, por cada pregunta se hizo un diagrama de barras o torras, y análisis frente a lo

encontrado en cada una de las empresas. Se buscó en cada empresa al menos dos de los siguientes roles: Analista de TI, Analista de Seguridad, Arquitecto o Auditor.

Los entregables de esta fase fueron: el diseño de la encuesta y los resultados de las entrevistas.

2.2. Fase 2: Determinar cuáles son las mejores prácticas para el manejo de la autorización en las organizaciones.

Para dar cumplimiento a esta fase se tuvo como referencia empresas reconocidas en el medio las cuales definen los modelos o mejores prácticas del manejo de la autorización en las aplicaciones con el fin de poder seleccionar las que ayuden a optimizar el costo, rendimiento, escalabilidad, seguridad, auditoria y lograr que solo las personas autorizadas realmente puedan acceder a la información en el momento adecuado. Estas empresas publican informes, modelos, diseños, herramientas, publicaciones, entre otros las cuales son un insumo fiable para cumplir con el objetivo. Se definió el criterio de que dicha documentación estuviera en el rango de años 2014 a 2020. Estas fuentes de información fueron Gartner, NIST, IEEE, OWASP, SANS, ICONTEC, MITRE, Researchgate y algunos libros relacionados con el tema. La documentación recolectada se analizó, se substrajo lo relevante que ayudara en la definición del modelo final. Así mismo, en la Tabla 2-1 se hizo una comparación entre los diferentes modelos indicando sus ventajas y desventajas de cada uno.

Tabla 2-1. Comparación entre modelos de autorización.

MODELO DE CONTROL DE ACCESO	VENTAJAS	DESVENTAJAS
	✓	–

También se buscó acerca de los estándares que se utilizan al momento de implementar el control de acceso en una aplicación y se hizo una tabla comparativa entre ellos indicando sus pro y contras, su formato se puede observar en la Tabla 2-2.

Tabla 2-2 Comparación entre los estándares de autorización.

ESTÁNDAR	VENTAJAS	DESVENTAJAS
	✓	–

El entregable en esta fase fue la identificación de 7 modelos, los estándares de control de acceso y sus respectivas comparaciones entre ellos.

2.3. Fase 3: Establecer los requerimientos mínimos requeridos para el control de acceso en las aplicaciones a la medida, en cuanto a la autorización.

De acuerdo con los resultados arrojados en las dos primeras fases, se seleccionaron los requerimientos mínimos necesarios en el modelo centralizado de autorizaciones que ayuden a las organizaciones a mitigar la fuga de información u otras amenazas por una mala implementación en el control de acceso. Estos requerimientos fueron modelados con la metodología ágil SCRUM y se siguieron los siguientes pasos [24]:

A. Definir el Backlog de producto, que corresponde al listado de necesidades para el diseño del modelo centralizado.
B. Asignar el tamaño a cada HU, para ello se utilizó la técnica de tallaje de camisetas donde: muy pequeña corresponde a XS, pequeña equivale a S, mediana es M, grande igual a L y se mide por la complejidad en el desarrollo que puede tener la HU, el valor es asignado por la experiencia del equipo de desarrollo.
C. Priorizar las HU con el fin de clasificarlas según el orden de importancia o urgencia. Se definieron 3 valores: 1) *Alto* equivale a las mínimas funcionalidades requeridas para poner en producción la aplicación; 2) Media son las funcionalidades que no son requeridas para funcionar pero que ayudan a optimizar el trabajo a los usuarios; y 3) Bajo corresponde a funcionalidades que no son prioritarias y pueden esperar para implementarse.

La plantilla utilizada para el backlog de producto fue definida con 5 columnas: Identificador único, Título de la HU, Funcionalidad a la que apoya, el tamaño y su prioridad. Esto se puede observar en la Tabla 2-3.

Tabla 2-3. Formato Backlog de producto

ID	Título	Funcionalidades	Tamaño	Prioridad
HU-XX				

Cada historia de usuario fue detalla apoyándose en las buenas prácticas mencionadas por los autores Abada y Salazar [24], en la siguiente tabla se puede observar la plantilla utilizada para la documentación.

Tabla 2-4. Formato Historia de Usuario

HISTORIA DE USUARIO	
Identificación: HU-XX	**Usuario**:
Título:	
Prioridad:	**Tamaño**:
Funcionalidad:	
Responsable:	
Descripción: **Como:** xxxx **Quiero:** xxxx **Para:** xxxx	
Criterio de aceptación: **CA1:** Título **Dado:** xxxxx **Cuando:** xxxxx **Entonces:** xxxx	

Para el diseño se utilizaron prototipos y algunos diagramas UML como entidad relación, casos de uso, interacción para modelar el proceso de control de acceso definido.

El entregable de esta fase fueron los requerimientos, el diseño del modelo centralizado de autorización, y la herramienta con la cual se hizo la prueba de concepto.

2.4. Fase 4: Validación el modelo de seguridad a través de una prueba de concepto, que permita establecer el nivel de cumplimiento en el control de acceso.

Tuvo como propósito la adecuación del ambiente tecnológico para construir la aplicación desarrollada a la medida, se definió el entorno de desarrollo a utilizar, el lenguaje de programación y el motor de base de datos. Es de aclarar, que no se encuentran definidas todas las historias de usuario con sus criterios de aceptación, solo se desarrolló lo mínimo requerido donde se pudiera validar el modelo con algunas entrevistas para verificar su utilidad y reducción de los problemas planteados al no disponer de un modelo centralizado de autorización para aplicaciones desarrollas a la medida.

Después de desarrollado el prototipo, se realizaron las pruebas de funcionamiento del producto mínimo viable para verificar correcto funcionamiento y poder posteriormente se llevó a cabo la presentación de la prueba de concepto por medio de entrevistas a mínimo 4 personas que manifestaron en el objetivo 1 que su control de acceso no es eficiente. Se hizo una presentación en PowerPoint donde se mostraron los principales diagramas del modelo definido, luego se hizo la presentación del prototipo y finalmente, los entrevistados diligenciaron la encuesta donde se evaluó dicho modelo.

El entregable en esta fase es la construcción del modelo con los requisitos mínimos de funcionamiento y el resultado de las entrevistas donde se evalúa este.

3. Resultados

A continuación, se presentan los resultados de cada una de las fases propuestas en la metodología.

3.1. Identificación de cómo manejan la autorización en las aplicaciones a la medida y su gobierno, en al menos cuatro organizaciones.

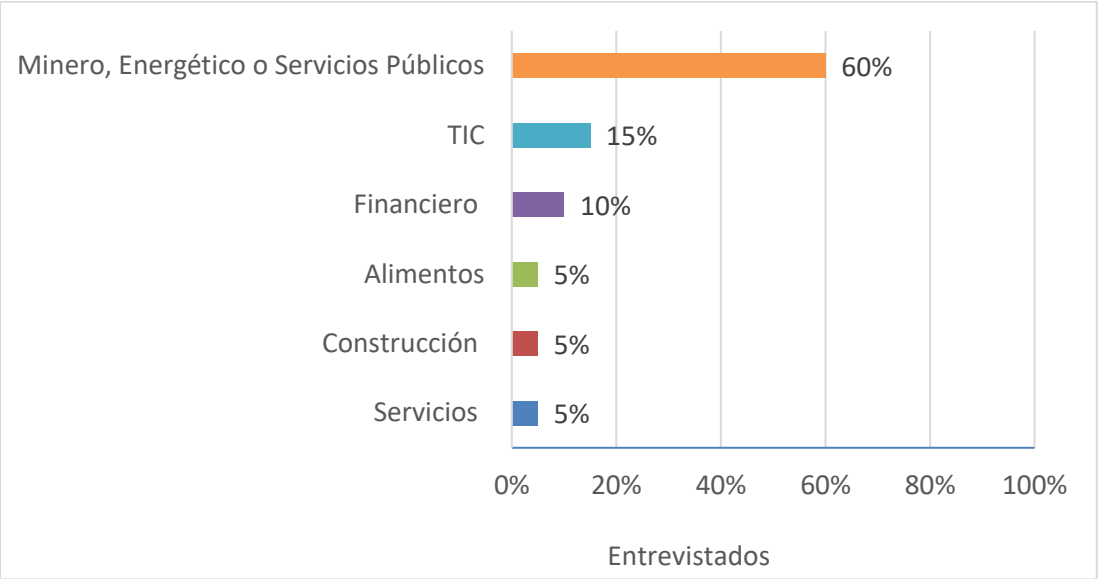
Los nombres de empresas y personas no fueron revelados en este documento por la Ley 1581 de 2012 de Protección de Datos Personales y porque la información de las encuestas es sensible de que un atacante pueda utilizarla para su beneficio.

Como se indicó en la metodología, se realizaron entrevistas a personal de TI en diferentes sectores empresariales y durante estás se diligenció la encuesta definida, con el fin de identificar el estado actual de las organizaciones frente al tema de autorización o control de acceso a las aplicaciones desarrolladas a la medida.

Se lograron realizar las entrevistas virtuales a 20 personas distribuidas en 11 empresas de varios sectores durante el mes de marzo del año 2020, donde el 60% de los entrevistados pertenecen al sector "Minero, energético o servicios públicos", el 15% son del sector TIC y el resto pertenecen a otros sectores, para ver más detalle ir a la Figura 3-1. Estas entrevistas no se pudieron realizar de manera presencial debido a que el país se encontraba en cuarentena durante el mes de marzo por causa del COVID 19 (Coronavirus), el cual fue reconocido por la Organización Mundial de la Salud (OMS) como una pandemia global el 11 de marzo de 2020 [3].

[3] https://www.who.int/news-room/detail/08-04-2020-statement-of-the-twenty-fourth-ihr-emergency-committee

Figura 3-1 Sector empresarial entrevistado.

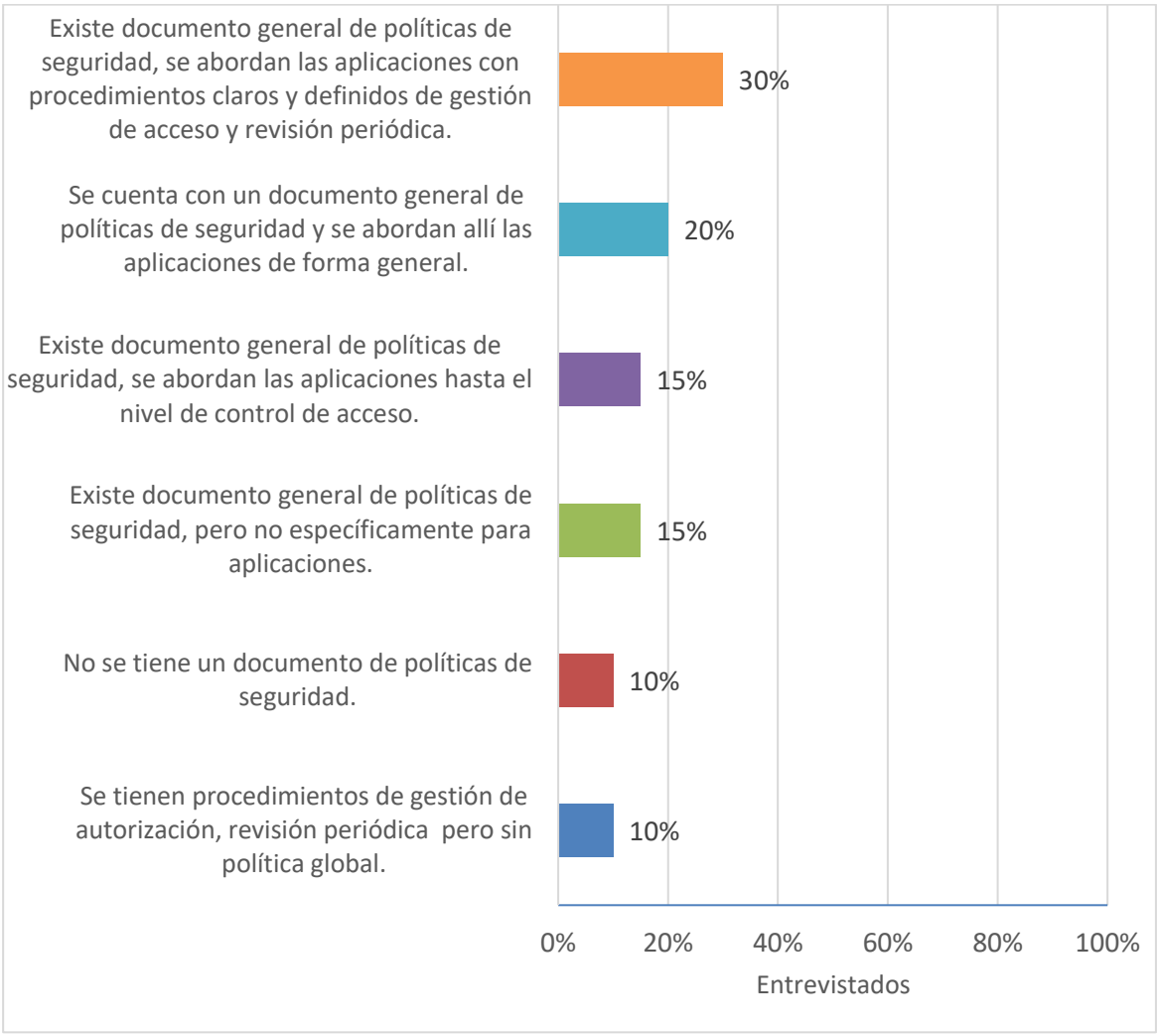


Fuente: Propia.

A continuación, se exponen los 14 resultados obtenidos por cada una de las preguntas evaluadas a los entrevistados:

a. Para la pregunta: ¿La empresa tiene definida alguna política de control de acceso o autorización a nivel corporativo para las aplicaciones desarrolladas a la medida o para paquetes comprados?, las respuestas de los entrevistados se pueden observar en la Figura 3-2. Se logró determinar que la mayoría de las empresas tiene un documento de política para el control de acceso a las aplicaciones, sin embargo, solo el 30% de las empresas respondieron que hacen su revisión periódica. El resto de las empresas no tienen un gobierno bien definido con respecto al control de acceso por lo que puede implicar un riesgo de seguridad frente al tema de la confidencialidad de la información.

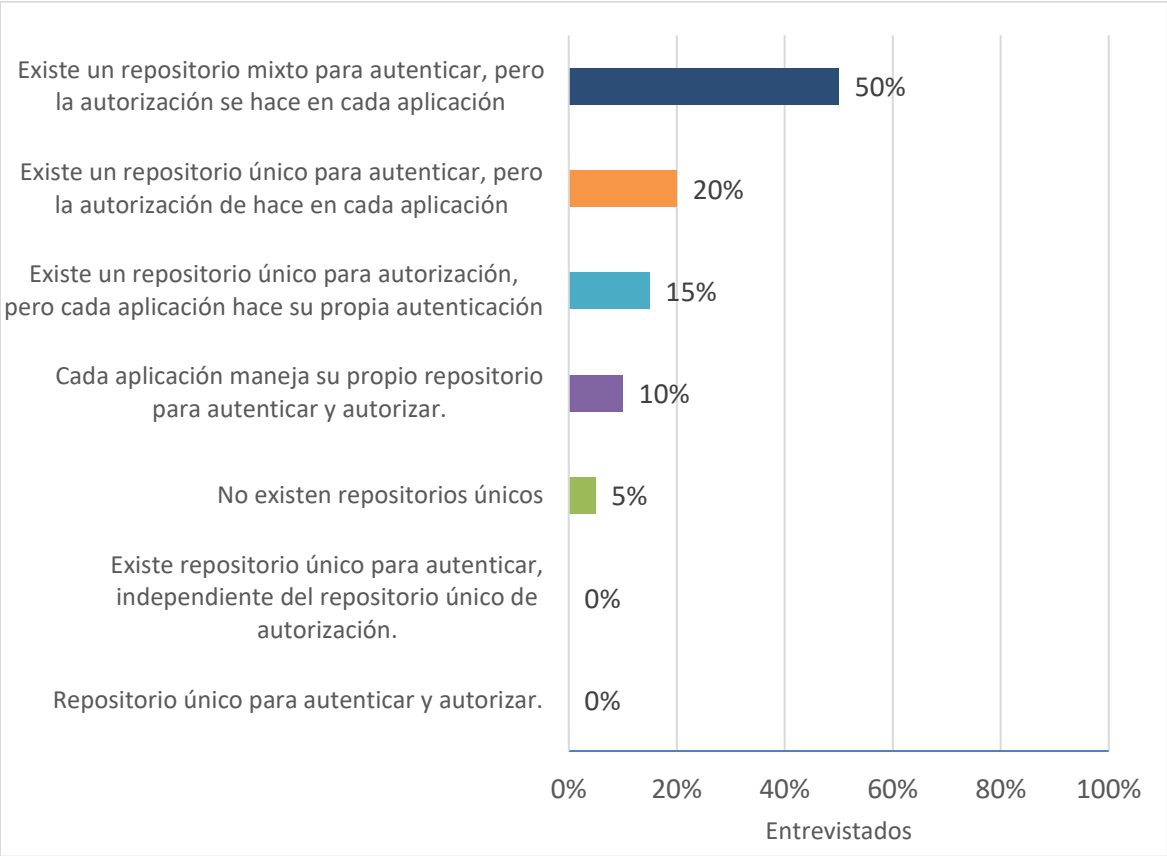
Figura 3-2. Política de control de acceso definida.



Fuente: Propia.

b. Frente a la inquietud: *¿La empresa maneja un único repositorio para la autenticación y la autorización o cada aplicación tiene su propio mecanismo?*, en las respuestas de los entrevistados, se puede observar que el 50% de las empresas utilizan un repositorio mixto de autenticación y un repositorio independiente de autorización por cada aplicación, y el 20% tiene un único repositorio de autenticación pero siguen las aplicaciones con su propio mecanismo de autorización, para más detalle ver la Figura 3-3. El utilizar diferentes repositorios de autenticación y autorización puede causar que se implemente incorrectamente el acceso a los recursos de un usuario. Así mismo, si cada aplicación tiene un repositorio de autorización, su implementación es más costosa y mayores los riesgos que un sistema centralizado.

Figura 3-3 Repositorio único o descentralizado para la autorización.

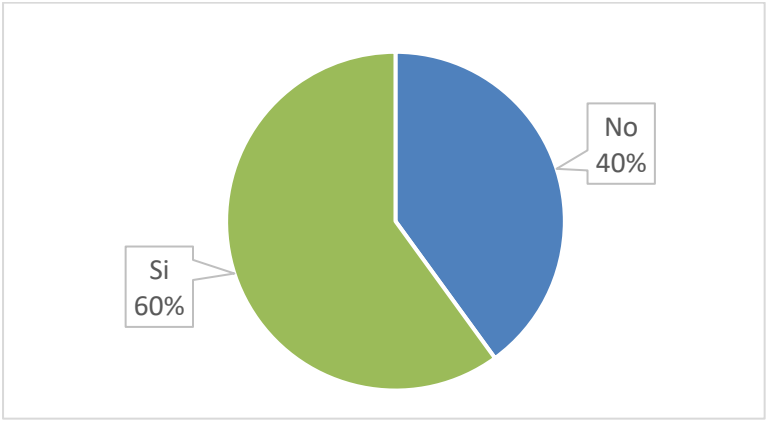


Fuente: Propia.

c. En la consulta: *¿Conoce alguna herramienta que se encargue de la gestión de acceso o autorización en las aplicaciones desarrolladas a la medida?* Las respuestas se pueden observar en la Figura 3-4. El 60% de los entrevistados conocen algunas herramientas para el apoyo de un modelo centralizado de autorización, entre los que mencionaron son:

- Oracle: SSO, Oracle identity manager u Oracle Identity Cloud Service.
- Directorio activo
- Desarrollos propios
- USM Administración Servicios unificado
- Centrify
- IAM Microsoft (Azure-Ad, Azure-B2C)
- Auth0
- Okta

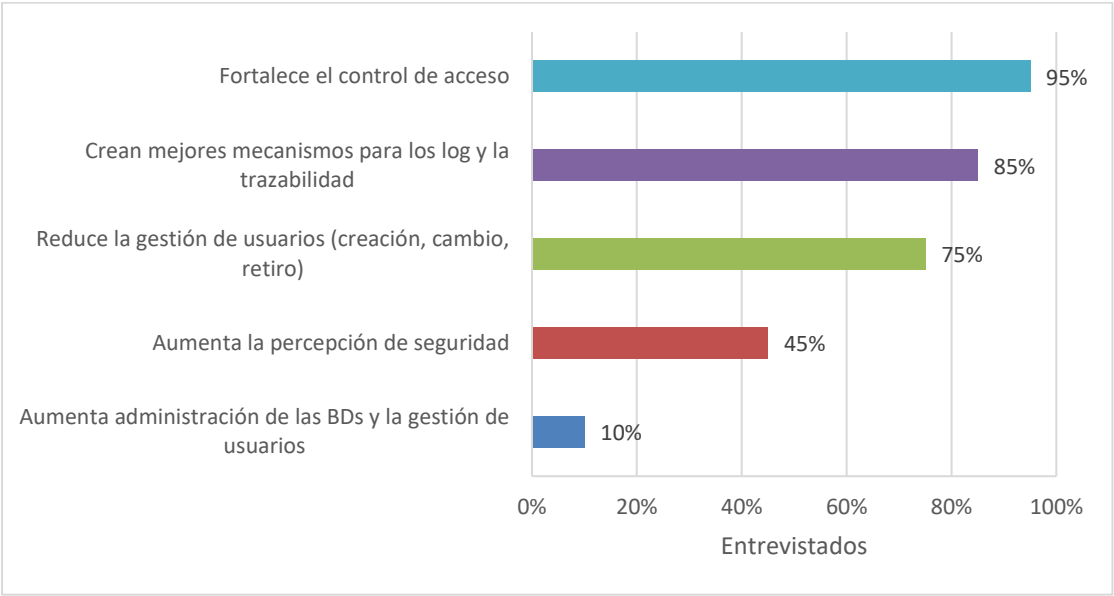
Figura 3-4 Herramientas para gestión de acceso.



Fuente: Propia.

d. Para la siguiente duda: *¿Qué elementos considera importante sobre un modelo centralizado para la administración de las autorizaciones en las aplicaciones?*, en la Figura 3-5 se pueden observar los resultados. Se concluyó que todos los entrevistados están de acuerdo con que se fortalece el control de acceso y se crean mejores mecanismos de log y trazabilidad. De hecho, la mayoría está de acuerdo con que no se aumenta la administración de la gestión de usuarios, solo el 10% respondió afirmativo frente al modelo de autorización centralizado.

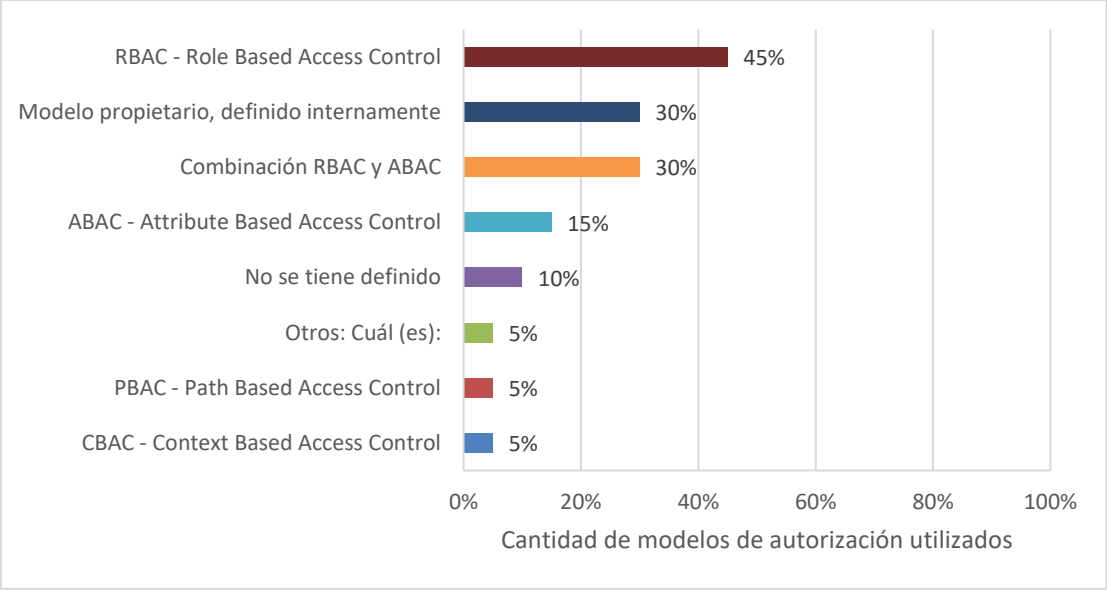
Figura 3-5 Elementos importantes sobre un modelo centralizado de autorización.



Fuente: Propia.

e. Con respecto a la cuestión: *¿Qué modelo de autorización utilizan las aplicaciones desarrolladas a la medida?* Según la Figura 3-6 los modelos más utilizados en las empresas son el RBAC, combinación del RBAC y ABAC y un modelo propietario, es decir, un desarrollo a la medida. Dentro de las razones de esta última, es porque en su investigación de mercados, ninguna herramienta cumplió el 100% de los requisitos definidos en la organización.

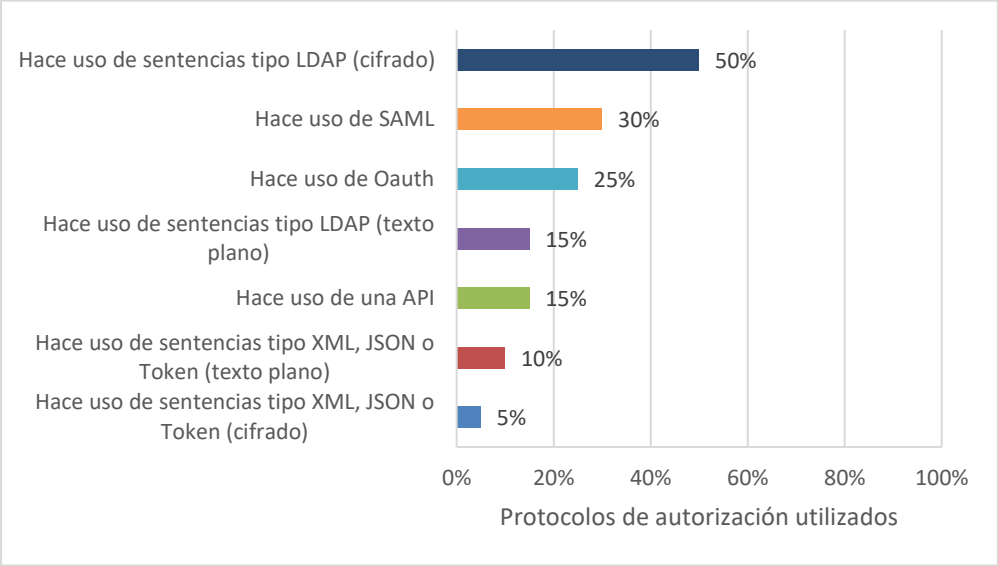
Figura 3-6 Modelos de autorización.



Fuente: Propia.

f. Frente a la pregunta: *En caso de tener una aplicación centralizada para la autorización, ¿cómo es el proceso realizado?* De acuerdo con la Figura 3-7, los modelos de autorización frecuentemente utilizados en las empresas son LDAP cifrado con un 50%, le sigue SAML con un 30% y el OAuth con un 25%.

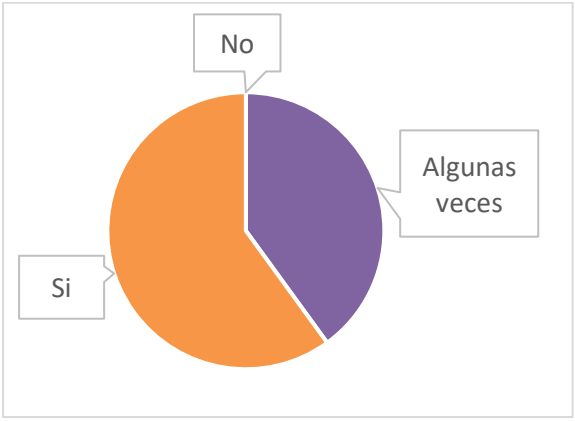
Figura 3-7 Modelos de autorización centralizados más utilizados.



Fuente: Propia.

g. Con respecto a la inquietud: *¿En la etapa de levantamiento de requisitos definen la gestión de acceso a la aplicación?* Como aparece en los resultados arrojados en la Figura 3-8, solo el 60% identifica los requisitos de control de acceso en las aplicaciones desde el inicio de un desarrollo de software. La especificación de la autorización es importante hacerla desde principio para evitar riesgos de seguridad como fuga de información o consulta de información por usuarios no autorizados.

Figura 3-8 Identificación de requisitos de autorización en las aplicaciones.



Fuente: Propia

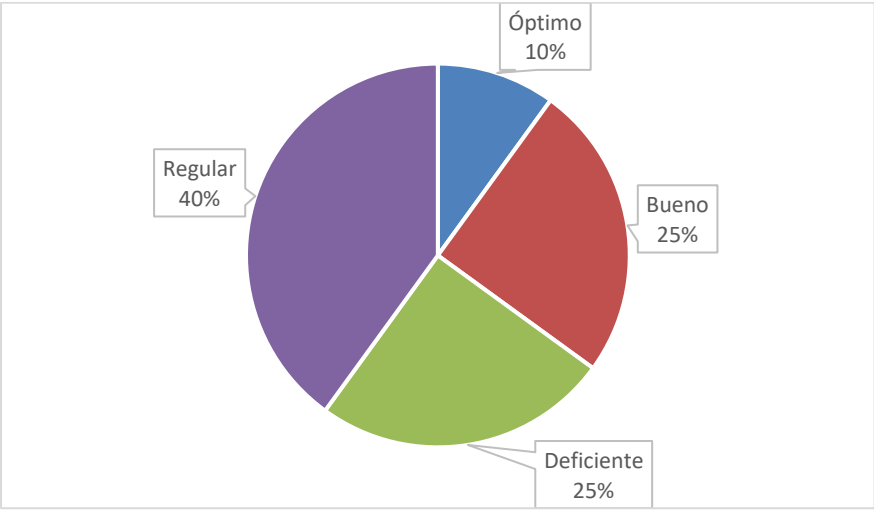
h. Para la duda: Si la aplicación maneja su propio modelo descentralizado de la autorización, ¿qué tan efectivo es la inactivación o modificación de los perfiles cuando una persona cambia de funciones o se retira de la empresa? Según la Figura 3-9, se logró determinar que el 40% mencionó que es regular y el 25% indicó que es deficiente el proceso de inactivación o modificación de la autorización en las aplicaciones.

Algunas de las razones que mencionaron los entrevistados del porque es deficiente o regular la inactivación o modificación de la autorización en las aplicaciones fueron:

- Al no existir una política, el proceso de notificación de cambio de rol es muy demorado y en ocasiones no se hace.
- Demoras por la oportunidad a la hora de inactivar o activar.
- No hay certeza que se quiten todos los permisos debido a que se hacen de manera independiente y por distintos grupos.
- Las integraciones para el desaprovisionamiento no están automatizadas.
- El administrador de la aplicación probablemente no se entere del cambio de funciones o el retiro de la persona por lo que la inactivación o modificación de los perfiles podría no darse.
- Es complicado porque muchas veces no se tiene el control de los usuarios que cambiaron de perfil, de rol, o usuarios no existentes en la empresa. Estos controles se llevan por BD y su depuración no está automática, depende del analista o administrador de BD que hace una depuración masiva de quien no hace uso de la aplicación y es cada cierto periodo lo cual dificulta tener al día este control.

Los entrevistados que mencionaron que era bueno u óptimo es porque manejan pedidos y se ejecutan flujos automáticos o manuales para cada aplicación debido a que sus empresas son pequeñas y permite realizar este tipo de acceso de manera centralizada y administrada por un usuario. Cuentan con un flujo de tareas que les permite la activación inactivación ingreso y retiro de cuentas de usuarios a las aplicaciones.

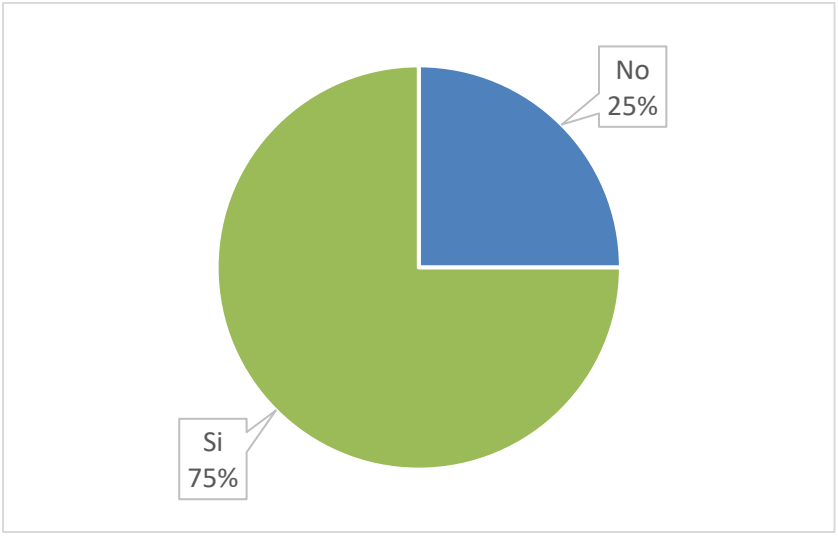
Figura 3-9 Efectividad en la inactivación o modificación del control de acceso.



Fuente: Propia.

i. En la consulta: *¿Las aplicaciones desarrolladas a la medida parten del principio del mínimo privilegio para la gestión de accesos?*, Como se muestra en la Figura 3-10, el 75% de los entrevistados respondieron que sí, esto es una buena práctica de seguridad, debido a que ejecuten código solo los permisos necesarios para completar las tareas requeridas y nada más y disminuyen los riesgos de seguridad en las aplicaciones.

Figura 3-10 Principio mínimo privilegio en las aplicaciones.



Fuente: Propia.

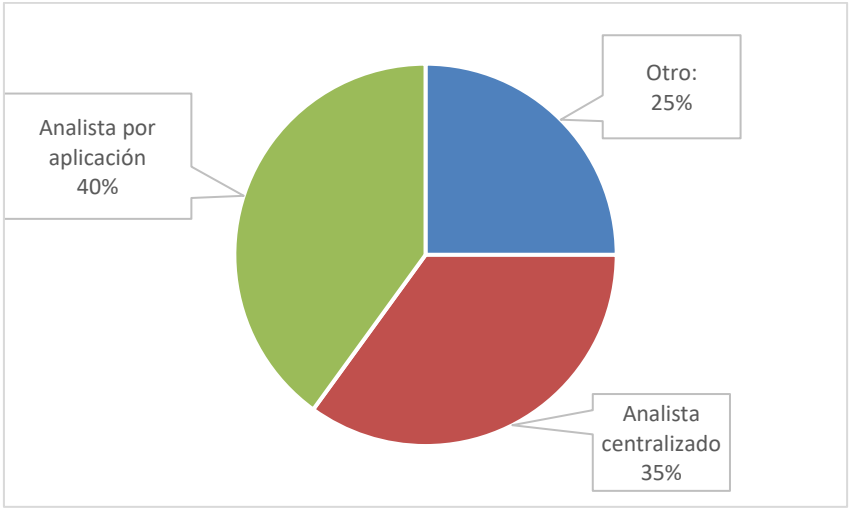
j. Frente a la pregunta: *¿La persona encargada de la seguridad en las aplicaciones desarrollas a la medida es una sola para todas o tienen analista de seguridad por aplicación?* Se puede observar en la Figura 3-11, el 40% lo realiza un analista de aplicación, el 35% lo hace un analista centralizado y el 25% restante lo realiza otro rol diferente.

Algunas de las respuestas de los entrevistados frente a otro rol diferente al mencionado son:

- Es manejado por cada área o proceso para el cual se desarrolla la aplicación.
- Se cuenta con un equipo dedicado a esa labor, dentro de cada equipo hay analistas por aplicación, pero todos están en capacidad de apoyar las diferentes aplicaciones dentro de su equipo.
- Es un área independiente de la empresa, con estructura jerárquica.
- Cada analista se encarga de la seguridad de la aplicación, pero no se tiene un proceso definido para indicar que son analistas de seguridad como tal.

En general, se puede concluir que existe un analista por aplicación encargado de otorgar o retirar los permisos en una aplicación, lo que corresponde a un modelo descentralizado y que se depende de una persona para atender esa labor, lo que podría ocasionar demoras en la atención y una persona podría continuar teniendo acceso a la información aunque sus funciones hayan cambiado.

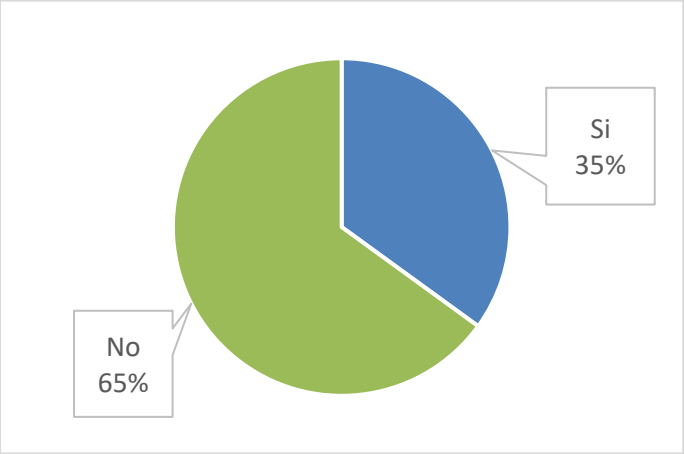
Figura 3-11 Analista de seguridad en las aplicaciones.



Fuente: Propia.

k. Para la cuestión: *¿Tiene implementado algún proceso automático de tal forma que al retirarse un empleado de la compañía, las aplicaciones eliminen o inactiven el usuario automáticamente, en vez de esperar a que lo realice una persona de forma manual?* De acuerdo con la Figura 3-12, el 65% de las empresas informaron no tienen un proceso automático para el retiro de usuarios en las aplicaciones, esta pegunta va muy alineada con la anterior donde se indica que existe un analista por aplicación encargado de realizar el proceso.

Figura 3-12 Proceso automático para retiro de usuarios en aplicaciones.



Fuente: Propia

l. Con respecto a la duda: *¿Tiene un promedio de cuántas solicitudes de gestión de acceso pueden llegar por mes? Creación, retiro, adición, reasignación de privilegios*. En la Tabla 3-1 se indicó por cada empresa su tamaño (Pequeño, Mediana o Grande) y la cantidad de solicitudes sobre la asignación o retiro de usuarios en las aplicaciones por mes. Como se observa en la tabla, entre más grande es la empresa el número de solicitudes incrementa por lo tanto es recomendado implementar un modelo centralizado del control de acceso para gestionar de manera oportuna cuando un usuario cambia de área o termina su contrato.

Tabla 3-1 Solicitudes de autorización en aplicaciones promedio por mes.

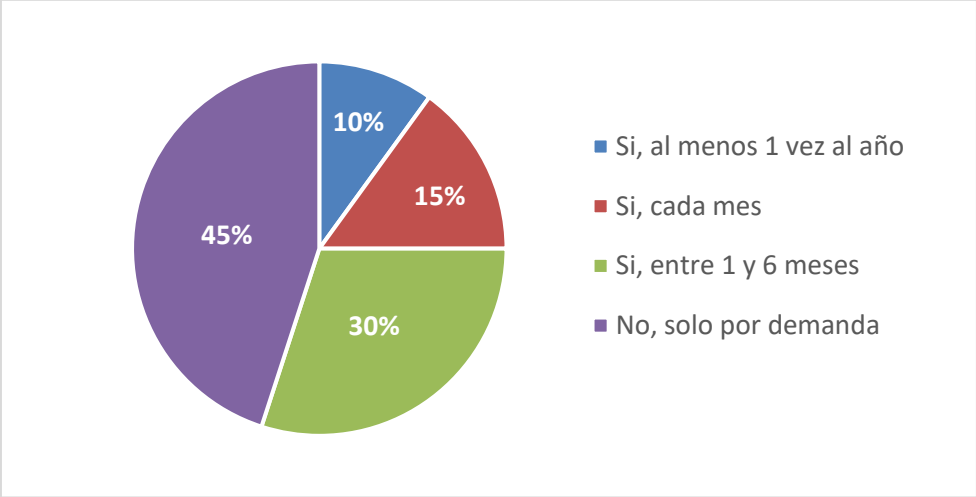
Empresa	Tipo de empresa	Cantidad por mes
1	Pequeña	3
2	Pequeña	10

Empresa	Tipo de empresa	Cantidad por mes
3	Pequeña	20
4	Pequeña	20
5	Pequeña	50
6	Mediana	60
7	Mediana	100
8	Mediana	200
9	Mediana	400
10	Grande	800
11	Grande	1000

Fuente: Propia.

m. En la consulta: ¿En la empresa realizan periódicamente la revisión de los permisos asignados a los usuarios en la aplicación? Se observa que en la Figura 3-13 solo el 45% de los entrevistados mencionaron que la revisión periódica de los permisos en las aplicaciones se hace es por demanda, es decir, alguien hace una solicitud para realizar dicho trabajo. El 30% hace la revisión entre 1 y 6 meses. En general, no es muy frecuente la revisión de estos permisos en las aplicaciones lo que puede ocasionar que personas no autorizadas puedan ver la información.

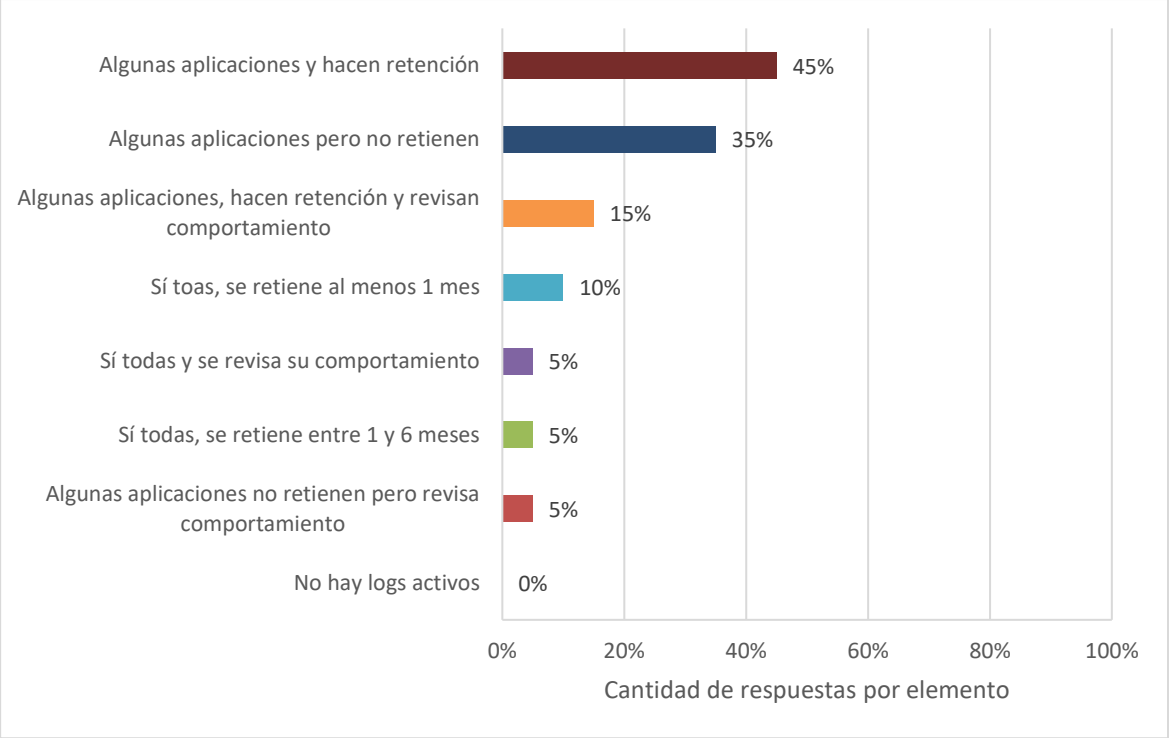
Figura 3-13 Revisión periódica de la autorización en las aplicaciones.



Fuente: Propia.

n. Para la pregunta: ¿Las aplicaciones cuentan con un log o registro de auditoria de control de acceso? Según la Figura 3-14, solo el 45% de las personas informaron que en temas de trazabilidad y auditoría se realiza a ciertas aplicaciones y se hace retención de estos archivos. El 35% respondieron que se hace para algunas aplicaciones, pero no se retiene los logs. En general, algunas de las empresas manejan un log por aplicación, pero no todas ni tampoco se monitorea por lo que un atacante podría ingresar al sistema consultar la información y no ser detectado a tiempo. De hecho mencionaban, que los logs se revisan de forma reactiva que proactiva, es decir, cuando hay un incidente es cuando se revisa realmente.

Figura 3-14 Manejo de auditoría en las aplicaciones.



Fuente: Propia.

Al realizar el análisis de los resultados de la encuesta se logró determinar que la mayoría de las empresas no tienen un modelo centralizado para la gestión del control de acceso, no son muy fuertes frente al tema de gobierno y no cuentan con un monitoreo del comportamiento de los usuarios en las aplicaciones. Con estos hallazgos se puede concluir que las empresas encuestadas pueden incurrir en una vulnerabilidad tal como modificación, eliminación o fuga de información

afectando la reputación y la economía de la empresa. Por lo anterior, las organizaciones requieren un componente centralizado para el manejo de la autorización en las aplicaciones, bien definido tanto desde la parte de Gobierno como desde la parte técnica, que asegure que solo los usuarios correctos pueden acceder en el momento indicado con sus privilegios definidos a las aplicaciones desarrolladas a la medida. Dicho componente forma parte de un sistema de Gestión de Identidades IAM (por sus siglas en inglés, Identity Access Management) el cual apoya el mejoramiento físico y digital en las organizaciones [42]. Con estos resultados arrojados por la encuesta se puede dar por cumplido el objetivo 1 de este proyecto.

3.2. Determinar cuáles son las mejores prácticas para el manejo de la autorización en las organizaciones.

Cuando la seguridad está incluida desde un principio en el ciclo de desarrollo de software se beneficia la protección del activo más importante para las organizaciones, como lo es la información. Si esto no ocurre, es dejarle la puerta abierta a un usuario malintencionado para que pueda realizar varios ataques como denegación de servicios, modificación, eliminación o fuga de datos, e incluso ver información que no está autorizado.

Con base en los resultados arrojados en la encuesta realizada a varias empresas, se concluye que el componente de autorización no es muy fuerte ni tampoco tienen un seguimiento continuo, lo que permite a un atacante tener éxito en violar los principios de confidencialidad e integridad de la seguridad de la información. Es por ello que, en este trabajo se buscaron los mejores modelos de control de acceso que ayuden a mitigar estos riesgos con el propósito de dar cumplimiento al objetivo 2 de este proyecto.

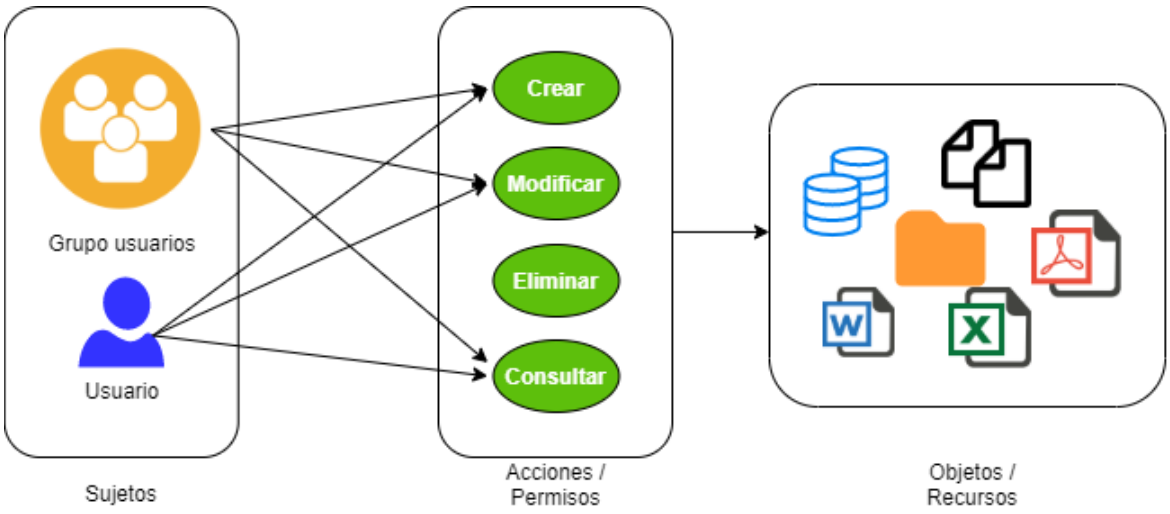
3.2.1. Modelos de autorización

A continuación, se presentan los resultados de la búsqueda realizada en la industria de las Tecnologías de la Información acerca de los modelos de control de acceso existentes para el desarrollo de aplicaciones, su definición y su respectivo diagrama de uso.

- **DAC - Discretionary Access Control**

El modelo de control de acceso discrecional es un mecanismo para restringir el acceso a la información en función de la identidad de los usuarios o por grupos definidos en la organización; Una de las herramientas que apoya este modelo es el Directorio Activo, el cual permite crear grupos y registrar usuarios. Basadas en el Directorio Activo, las aplicaciones implementan su control de acceso, de forma que, si el usuario pertenece a un grupo específico, podrá realizar las acciones esperadas sobre los objetos o recursos de la aplicación. Es un modelo donde el responsable del objeto se encarga de otorgar o retirar los permisos. Es común con el sistema operativo Windows cuando se comparten carpetas. Una acción o permiso se define como las opciones que tiene el sujeto o usuario en la aplicación, por ejemplo: crear, modificar, eliminar o consulta. Así mismo, se entiende por recurso u objeto: datos, servicios, aplicaciones, ejecutables, archivos, entre otros. Esto se puede observar en la Figura 3-15 [47][42].

Figura 3-15. Ejemplo Modelo DAC.



Fuente: Propia

- **RBAC - Role Based Access Control**

La sigla RBAC corresponde en español a *Control de Acceso Basado en Roles*. Es uno de los modelos de control de acceso más usado en las organizaciones, como lo demuestra la encuesta realizada a diferentes empresas. El acceso a las aplicaciones se basa en roles y permisos de un usuario o grupo

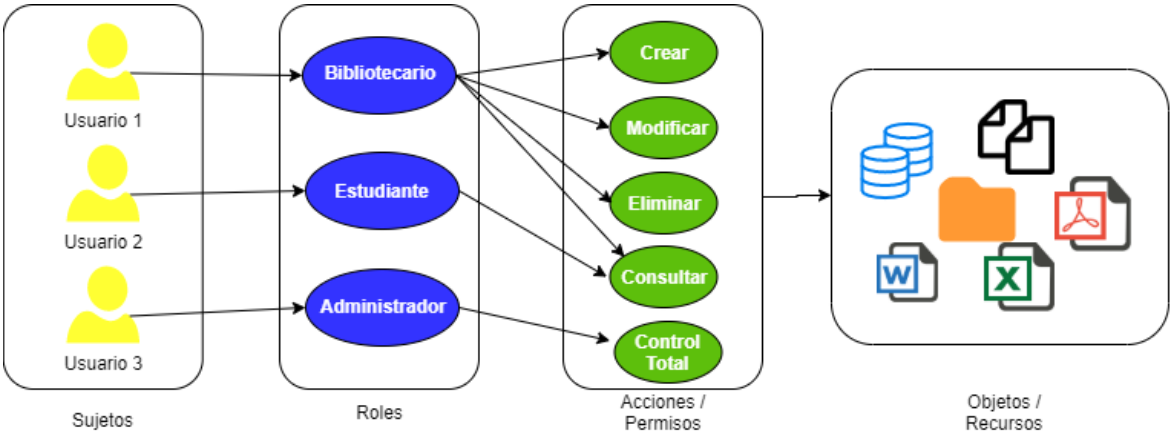
de usuarios en la organización. El permiso define la operación que puede realizar sobre el objeto o recurso. Se entiende por rol alguna función o responsabilidad dentro de una organización. La combinación del rol con los permisos define el concepto de RBAC. Un usuario puede tener uno o varios roles asignados y un rol puede tener uno o varios permisos o acciones. En este modelo, los permisos no se asignan directamente al sujeto sino al rol, lo que disminuye el soporte en la asignación de permisos y su control; sin embargo, puede existir un exceso de roles al aumentarse los recursos [47][43].

A continuación, se detallan los elementos que componen este modelo [47][43]:

- Sujeto: corresponde al usuario que desea tener acceso al recurso.
- Objeto: es el recurso al que el usuario desea tener acceso, este puede ser una base de datos, un archivo, un dato, entre otros.
- Rol: Es la función designada al usuario, generalmente corresponde al cargo que tiene definido en la organización.
- Acción: Es la operación que puede realizar el sujeto sobre el objeto.

En la siguiente Figura 3-16 se puede observar un ejemplo de un sistema de bibliotecas donde se definieron los siguientes roles: Bibliotecario, Estudiante y Administrador. Las acciones o permisos son: Crear, Modificar, Eliminar, Consultar y Controlar totalmente el sistema. El usuario 1 se identifica con el rol Bibliotecario el cual tiene las acciones o permisos de crear, modificar, eliminar y consultar libros. El Usuario 2 con el rol de estudiante solo puede consultar los libros; y por último, el Usuario 3 tiene el rol Administrador con el permiso de control total, es decir, que tiene acceso sin restricción a las configuraciones del software y también se encarga del soporte en caso de alguna falla en la aplicación.

Figura 3-16. Ejemplo Modelo RBAC.



Fuente: Propia

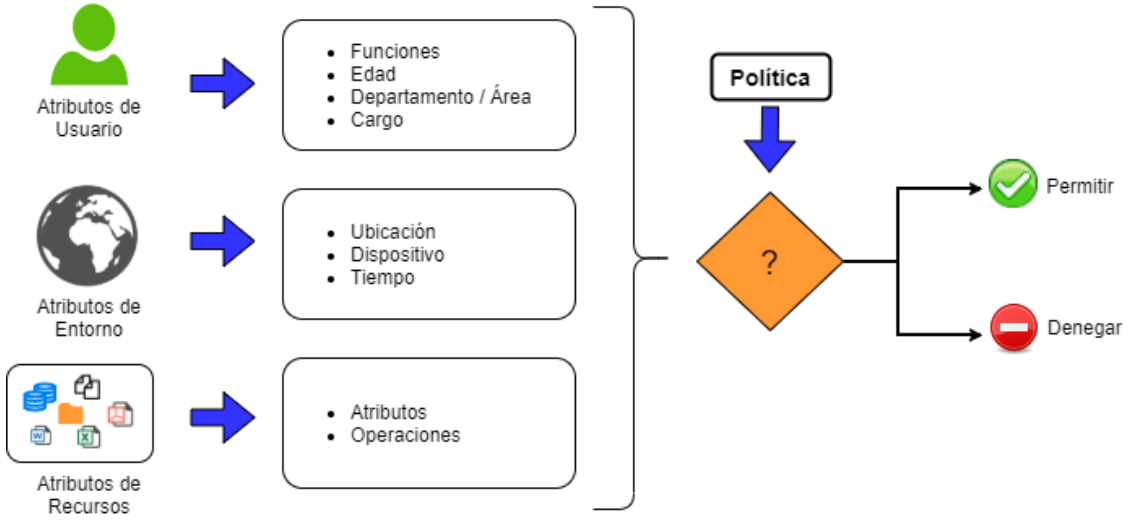
Este modelo se apoya en tres principios de seguridad [22]:

- Principio del mínimo privilegio: corresponde a los permisos mínimos necesarios con los que el usuario puede realizar sus funciones en la aplicación.
- Separación de funciones: permite definir roles excluyentes entre sí para evitar ser juez y parte al completar una tarea delicada dentro de la aplicación.
- Abstracción de datos: se pueden definir permisos además de las básicas en la aplicación según la necesidad del negocio.

- **ABAC - Attribute Based Access Control**

El modelo de Control de Acceso Basado en Atributos tiene como principales características que es flexible y escalable. Los permisos en la aplicación son asignados con relación a unos atributos o propiedades que identifican al usuario. Este enfoque es más granular y se basa en la evaluación de los atributos del usuario, las condiciones del entorno, los atributos del recurso y las políticas de control de acceso definidas. Este modelo ayuda a resolver el inconveniente con el RBAC relacionado con la explosión de roles creados. La organización debe definir los atributos para los sujetos, objetos y recursos. El objeto debe tener al menos una regla de control de acceso, un ejemplo de esto se puede observar en la Figura 3-17 [17],[42],[48].

Figura 3-17. Ejemplo Modelo ABAC.



Fuente: Propia

A continuación, se detallan los elementos que componente a este modelo [17]:

- Atributos de usuario: corresponde a las propiedades del usuario desde el punto de vista organizacional; pueden ser obtenidas de varias fuentes como recursos humanos, seguridad, entre otros. Algunos atributos son: funciones, edad, cargo, área a la que pertenece.
- Atributos del objeto: se refiere a las propiedades del recurso a las que el usuario desea acceder y a las acciones u operaciones que puede realizar sobre este, por ejemplo: crear, modificar, eliminar, entre otros.
- Condiciones del entorno: esta describe el contexto de la solicitud del control de acceso, no se asocia ni al sujeto ni al objeto; por mencionar: la fecha, hora, ubicación, dispositivo, entre otros atributos.
- Políticas: corresponde a las reglas de control de acceso y la combinación de los tres elementos mencionados anteriormente para permitir o denegar al usuario el acceso al recurso. Es importante disponer de un buen gobierno para el manejo de estas políticas.

Este modelo se apoya en tres principios de seguridad [18]:

- Principio del mínimo privilegio: corresponde a los permisos mínimos necesarios con los que el usuario puede realizar sus funciones en la aplicación.

- Separación de funciones: permite definir roles excluyentes entre sí para evitar ser juez y parte al completar una tarea delicada dentro de la aplicación.

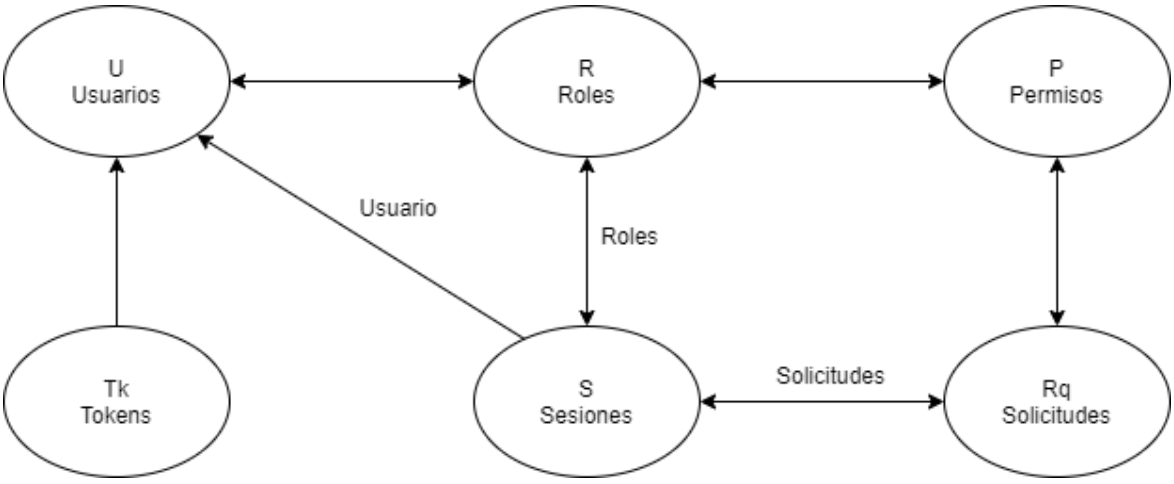
- **PBAC - Path Based Access Control**

El modelo de control de acceso basado en rutas permite a las aplicaciones un control de acceso flexible utilizando la ruta de la solicitud (URI - *Uniform Resource Identifier*), es una extensión al modelo RBAC debido a que se utilizan los elementos: usuarios, roles y permisos, y se adicionan sesiones, un token y la solicitud [15]. A continuación, su definición.

- Sesión: es un mapeo entre el usuario y su conjunto de roles asignados.
- Token: atributos de usuario que le permiten autenticarse en la aplicación.
- Solicitud (Rq – *Request*): es la petición que se envía del cliente al servidor, la petición tiene encabezados (*header*), URI y parámetros. La solicitud puede tener varias sesiones.

En la Figura 3-18 se puede observar los elementos que componen el modelo y sus relaciones entre ellos. Un usuario tiene los atributos de token, sesiones y roles. Las sesiones tienen solicitudes y roles. Los permisos son asignados a los roles y las solicitudes. Con la combinación de token, sesión, solicitudes, roles y permisos, la aplicación podrá definir el acceso o no del usuario en la aplicación.

Figura 3-18. Elementos del modelo PBAC.



Fuente: [15]

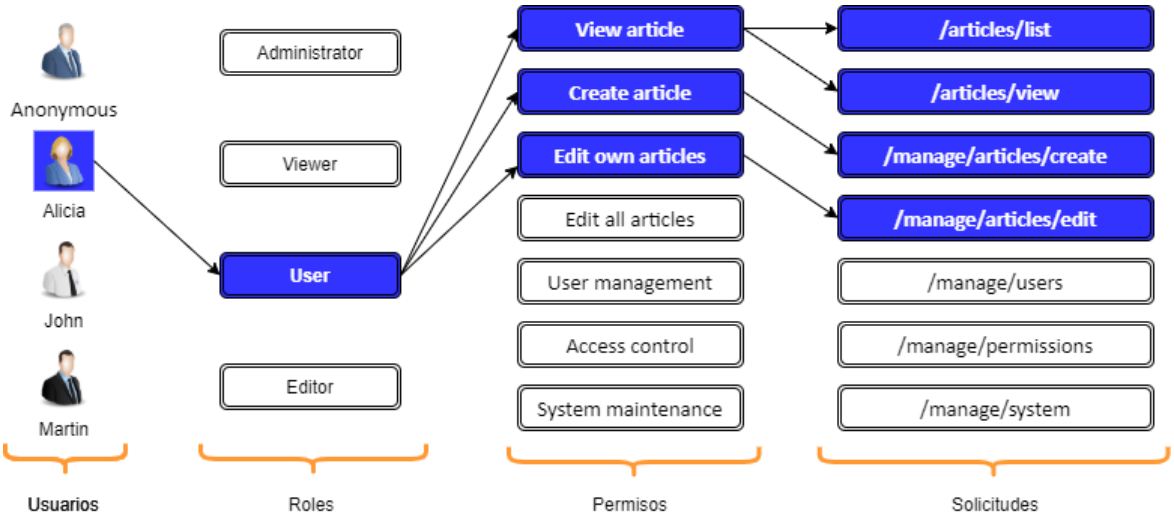
Un ejemplo de este modelo sería:

- Usuarios (U):
  - Anonymous,
  - Alice,
  - John,
  - Martin
- Roles (R):
  - Viewer,
  - User,
  - Editor,
  - Administrator
- Permisos (P):
  - "view article",
  - "create article",
  - "edit own articles",
  - "edit all articles",
  - "user management",
  - "access control",
  - "system maintenance"
- Solicitudes (Rq):
  - /articles/list,
  - /articles/view,
  - /manage/articles/list,
  - /manage/articles/create,
  - /manage/articles/edit,
  - /manage/users/list,
  - /manage/users/create,
  - /manage/users/edit,
  - /manage/permissions/roles,
  - /manage/permissions/acl,
  - /manage/system/settings,
  - /manage/system/maintenance

- Roles asignados a los usuarios:
    - Anonymous: Viewer
    - Alice: User
    - John: Editor
    - Martin: Editor, Administrator
- Permisos asignados a los roles:
    - Viewer: "view article"
    - User: "view article", "create article", "edit own article"
    - Editor: "view article", "create article", "edit all articles"
    - Administrator: "user management", "access control", "system maintenance"
- Asignación de las solicitudes a los permisos:
    - "view article":/articles/list, /articles/view
    - "create article": /manage/articles/create
    - "edit own article": /manage/articles/edit
    - "edit all article": /manage/articles/edit
    - "user management": /manage/users
    - "access control": /manage/permissions
    - "system maintenance": /manage/system

Con base en la definición anterior, el usuario Alicia tendría los siguientes accesos definidos en la aplicación, esto se puede observar en la Figura 3-19:

Figura 3-19. Ejemplo permisos modelo PBAC para un usuario.



Fuente: Propia

Este modelo se apoya en los siguientes principios de seguridad[15]

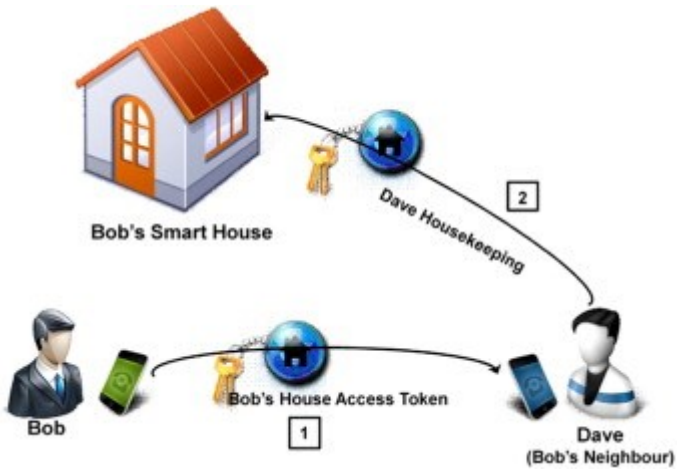
- Control de acceso centralizado: el proceso se realiza en un solo modelo sin delegarse a otros.
- Principio del mínimo privilegio: corresponde a los permisos mínimos necesarios con los que el usuario puede realizar sus funciones en la aplicación.
- Separación de funciones: las tareas con transacciones sensibles pueden requerir de varios usuarios para realizar la operación.
- Abstracción de datos: se pueden definir permisos a parte de las básicas en la aplicación según la necesidad del negocio.
- Separación por URI: Dividir el sistema en componentes separados accediendo por el identificador único del recurso (URI).

- **CapBAC - Capability Based Access Control**

En el modelo de control de acceso basado en capacidades los derechos de acceso se otorgan a los sujetos basados en el concepto de capacidad, que es una señal de autoridad transferible e inolvidable (por ejemplo, una clave y un ticket), y describe un conjunto de derechos de acceso para cada sujeto. Los marcos de autorización RBAC y ABAC no soportan modelos distribuidos, es por ello que surge este nuevo estándar para apoyar a las aplicaciones IoT (Internet de las cosas). En la Figura

3-20 se muestra un ejemplo del modelo CapBAC donde Bob le entrega un token de acceso a Dave para que este pueda ingresar a la casa y realizar la tarea de limpieza asignada. El token identifica que solo Dave tiene el acceso, lo que realmente puede hacer dentro de la casa y la vigencia de este. Bob y Dave no necesitan tener una relación de confianza con los componentes Autenticación y Autorización, así mismo, el token es solo dado para Dave, él no puede transferirlo a otra persona ni tampoco realizar otra actividad que no esté definida [49].

Figura 3-20. Ejemplo Modelo CapBAC.



Fuente: [49]

Los elementos que componente a este modelo son [49]:

- Recurso u Objetos.
- Sujeto o Usuarios.
- Los derechos otorgados: es la acción o permiso que puede ejecutar el sujeto.
- Cadena de autorización: La persona encargada de indicar el derecho otorgado al sujeto.
- Revocación de la capacidad: duración del permiso asignado al sujeto.

Este modelo soporta las capacidades[49]:

- Soporte de delegación: un sujeto puede otorgar permisos a otro sujeto.
- Revocación de capacidades: los permisos solo pueden ser retirados por personal autorizado.
- Granularidad de la información: permite definir el cambio dinámico en el control de acceso.

- Representación XML: Los tokens de capacidad están en archivos XML firmados.
- Basado en SAML / XACML.

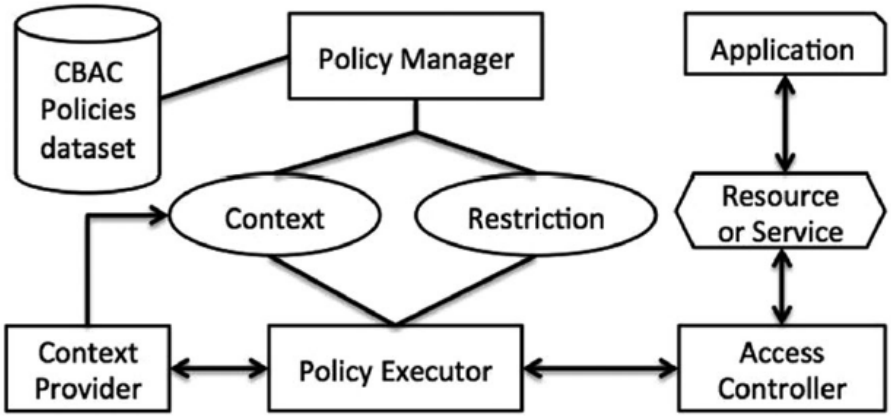
La delegación de derechos y el control de acceso basado en capacidades apoya los siguientes principios de seguridad[49]:

- Principio de la Autoridad Mínima (Privilegio mínimo): corresponde a los permisos mínimos necesarios con los que el usuario puede realizar sus funciones en la aplicación

- **CBAC - Context Based Access Control**

El control de acceso basado en el contexto (CBAC) otorga o niega el permiso basado en los atributos contextuales del sujeto que intenta ingresar al recurso. Su uso frecuente es en dispositivos de red, firewalls, dispositivos móviles. Con estas políticas se puede indicar cuándo y dónde el sujeto puede acceder a los recursos, reduciendo el vector de ataque para que personas malintencionadas puedan robar datos. Algunos atributos de contexto son: tipo de dispositivo (móvil, portátil, Tablet, entre otros), ubicación, fecha y hora. Los privilegios se otorgan o deniegan de manera automática, según el contexto de la petición del sujeto [50]. En la Figura 3-21 se puede apreciar el diseño del modelo para un dispositivo móvil.

Figura 3-21. Ejemplo Modelo Context BAC para dispositivos móviles.



Fuente: [50]

Los elementos que componente a este modelo para dispositivos móviles son [50]:

- Proveedor de contexto (CP – Context Provider): Es el encargo de obtener la ubicación del sujeto.
- Controlador de acceso (AC – Access Controller): Su función es controlar las autorizaciones en las aplicaciones, es decir, otorga o deniega el acceso al sujeto.
- Administrador de la política (PM Policy Manager): Es la interfaz donde se administran las políticas de acceso, es decir, allí se crean o eliminan éstas.
- Ejecutor de políticas (PE Policy Executor): Se encarga de ejecutar las políticas definidas al sujeto que está haciendo la solicitud.

- **Control de acceso basado en Blockchain**

Blockchain se define como una cadena de bloques que no se pueden modificar después de unirse y se utiliza para realizar transacciones o contratos de valor. Es como un libro contable distribuido en la red; cada nodo ubicado sobre la red tiene una copia exacta del bloque y las transacciones son aprobadas por los participantes en la red. Cada bloque está unido por un código único que se obtiene del hash del bloque anterior, si este no coincide, no puede unirse. Para poder unirse a la cadena de bloques, uno de los nodos debe responder un desafío o prueba de trabajo, si es correcto, se le permite agregar el bloque y obtiene un beneficio o bonificación. Hay dos tipos de Blockchain: 1) Públicas: son de acceso público y cualquier persona con un equipo puede pertenecer a la red; 2) Privadas: su uso es exclusivo y para pertenecer a la red debe ser autorizado. Uno de los primeros usos de esta tecnología fue con BitCoins. El uso de la tecnología Blockchain no es solo para bitcoins, su filosofía también se puede aplicar en otros campos de acción como el sector gobierno, educativo, salud, empresarial, entre otros. Por ejemplo para la creación y consulta de registros civiles, de propiedad, historia clínica, generación de diplomas educativos [51][52].

Se entiende por contrato inteligente (Smart contract) un conjunto de instrucciones u operaciones que tienen una validez indefinida y no pueden modificarse debido a que utilizan la tecnología Blockchain, generando transparencia e integridad a los contratos. Un ejemplo de esto lo podemos encontrar en el libro Blockchain de los autores Jesús Díaz, Luis Hernández y David Arroyo, el cual fue citado textualmente a continuación [53]:

"Bloquear un coche en el caso de que su propietario no satisfaga su mensualidad del préstamo concedido por un banco. Si fuera así, entonces:

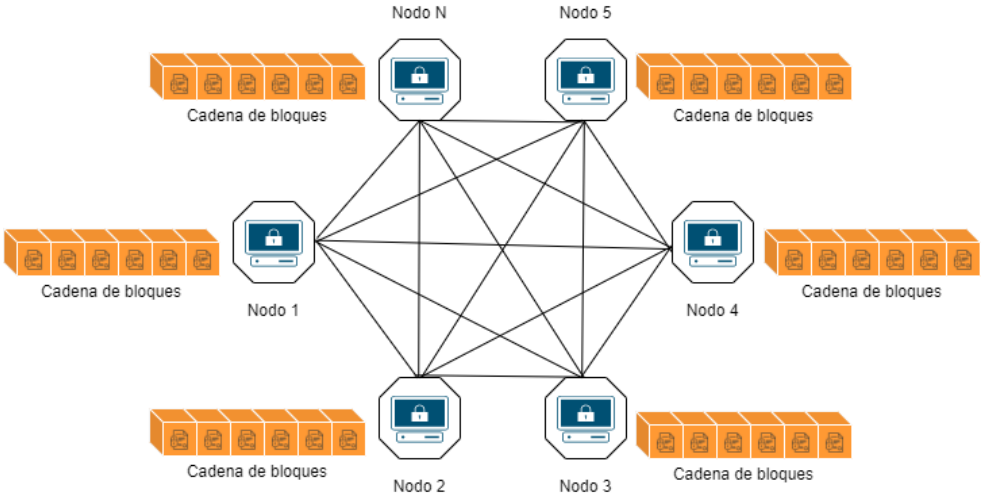
- El propietario ejecuta la operación de apertura del coche. Esta operación llega a todos los ordenadores de la red a través del coche, que es uno de estos ordenadores.
- Cada ordenador comprueba el registro inmutable que el propietario no ha satisfecho el último pago. El resultado de la operación de apertura es denegado.
- Dicha denegación se escribe en el registro por parte de alguno de los ordenadores que componen la red.
- Al detectar la escritura del registro, el coche tiene la certeza de que el resultado es válido. Rechaza abrirse, informando al propietario del motivo."

Todo el caso anterior se encuentra firmado digitalmente con lo cual se puede identificar quién lo generó; al estar el mismo contrato en todos los nodos, es difícil modificarlo porque las cadenas de bloques se unen por el hash del bloque anterior, lo que no facilita a un atacante poder realizar algún cambio o eliminarlo porque debe hacerlo en todos los nodos y la tecnología Blockchain apunta a los principios de disponibilidad, confidencialidad e integridad de la seguridad de la información.

Los elementos que componen este modelo se describen a continuación y se pueden observar en la Figura 3-22:

- Cadena de bloques: cadena infinita que contiene la información.
- Nodos: los equipos donde se guardan las cadenas de bloque.

Figura 3-22. Ejemplo de una red Blockchain.

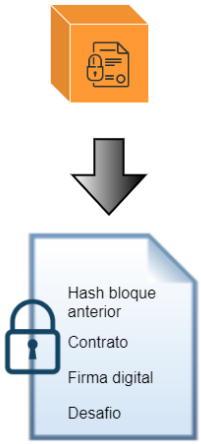


Fuente: Propia.

Cada cadena de bloque está compuesta por los siguientes atributos y se pueden observar en la Figura 3-23:

- Hash: para conocer la integridad del bloque anterior
- Desafío o prueba de trabajo: el acertijo que debe responder un nodo para poderse unir a la cadena de bloques.
- Contrato inteligente: las condiciones definidas por el proceso o empresa.
- Firma digital: indica quién firmó el contrato.

Figura 3-23. Estructura de una cadena de bloques.



Fuente: Propia.

3.2.2. Cuadro comparativo de modelos de autorización

En la Tabla 3-2 se describen las ventajas y desventajas de los modelos de control de acceso definidos anteriormente [47], [48], [42], [50] con el propósito de extraer de algunos las mejores prácticas y definir el modelo centralizado de autorización para aplicaciones desarrolladas a la medida.

Tabla 3-2. Cuadro comparativo modelos de autorización.

MODELO DE CONTROL DE ACCESO	VENTAJAS	DESVENTAJAS
DAC	✓ Facilidad de uso en las aplicaciones desarrolladas a la medida. ✓ Facilidad de administración desde el punto de vista de la aplicación. ✓ La autenticación y la autorización están unidas. ✓ Un único repositorio en toda la organización. ✓ Parte del principio de privilegios mínimos. ✓ El propietario del objeto tiene control total sobre el acceso otorgado. ✓ Los usuarios pueden pertenecer a varios grupos.	– La documentación de los roles y accesos debe estar definida y actualizada periódicamente. – Se puede producir un aumento de alcance, por ejemplo, se pueden otorgar más accesos y privilegios de lo previsto. – Limitación técnica del Directorio Activo porque un usuario puede estar en muchos grupos y se supere el tamaño del token definido por defecto. – Se requiere desarrollo desde la aplicación para identificar los grupos creados y las personas asignadas. – Modelo descentralizado.

MODELO DE CONTROL DE ACCESO	VENTAJAS	DESVENTAJAS
RBAC	✓ Los roles se crean con base en las funciones de los usuarios en la organización. ✓ Facilidad de uso en las aplicaciones para las aplicaciones desarrolladas a la medida. ✓ Facilidad de administración por parte de los usuarios encargados de la aplicación. ✓ Apoya a la segregación de funciones y el mínimo privilegio con el fin de evitar fraude o abuso de privilegios. ✓ Modelo centralizado. ✓ Los usuarios pueden pertenecer a varios grupos.	– La documentación de los roles y accesos debe estar definida y actualizada periódicamente. – Tendencia a crearse nuevos roles de los necesarios otorgando más permisos de los requeridos por el usuario. – Los roles son asignados de manera estática lo que carece de flexibilidad y sensibilidad. – Falta de escalabilidad para dominios federados. – No existe un mecanismo para diferenciar los usuarios y sus roles. – Contexto del usuario es estático. – Aunque apoya a la segregación de funciones, si la aplicación no tiene un gestor de reglas donde se definan las restricciones entre los roles, se puede llegar a incumplir este principio.

MODELO DE CONTROL DE ACCESO	VENTAJAS	DESVENTAJAS
ABAC	✓ Modelo centralizado. ✓ Es un modelo escalable. ✓ Las políticas son escritas usando un lenguaje entendible (XACML). ✓ Contexto del usuario es en tiempo real. ✓ Apoya a la segregación de funciones y el mínimo privilegio con el fin de evitar fraude o abuso de privilegios.	– Dificultad para definir los atributos en sistemas complejos. – La seguridad depende del número de atributos definidos, lo que puede afectar de manera negativa el rendimiento de la aplicación. – Se aumenta la complejidad en el mantenimiento de las reglas de control de acceso. – Su implementación es más costosa frente al modelo RBAC porque se requiere más nivel de detalle durante su codificación.
PBAC	✓ Facilidad de uso en las aplicaciones. ✓ Facilidad de administración. ✓ Apoya a la segregación de funciones y el mínimo privilegio. ✓ Modelo centralizado. ✓ Separación por URI.	– Se aumenta la complejidad en el mantenimiento de las reglas de control de acceso. – Contexto del usuario es estático.
CapBAC	✓ El Principio de la Autoridad Mínima (PoLA) (Privilegio Mínimo) es el predeterminado.	– Modelo descentralizado. – Requiere la emisión de capacidades para todos los sujetos.

MODELO DE CONTROL DE ACCESO	VENTAJAS	DESVENTAJAS
	✓ Admite un control de acceso más detallado. ✓ Tiene menos problemas de seguridad. ✓ Externaliza y distribuye la gestión del proceso de autorización. ✓ No necesita gestionar problemas relacionados con la complejidad y la dinámica de las identidades de los sujetos.	– Dificultad para conocer quién puede acceder a un recurso. – Dificultad para revocar todos los permisos para un recurso.
Context BAC	✓ Escalable. ✓ Contexto del usuario es en tiempo real. ✓ Apoya a la segregación de funciones y el mínimo privilegio con el fin de evitar fraude o abuso de privilegios.	– Dificultad para conocer quién puede acceder a un recurso. – Dificultad para revocar todos los permisos para un recurso. – Requiere buen conocimiento del contexto para poder otorgar o denegar los accesos. – La ubicación del dispositivo debe ser muy exacta para poder aplicar las políticas
Control de acceso basado en Blockchain	✓ Es difícil que ocurre fuga de privacidad porque no existen intermediarios. ✓ No tiene un punto único de falla.	– Modelo descentralizado. – Los nodos deben tener bastante capacidad de procesamiento para resolver el desafío más rápido que los demás nodos.

MODELO DE CONTROL DE ACCESO	VENTAJAS	DESVENTAJAS
	✓ Utilizar certificados digitales para garantizar no repudio y autenticidad. ✓ Algoritmos de hash fuertes.	– Los nodos deben tener bastante capacidad de almacenamiento para guardar las cadenas de bloques. – El tiempo de respuesta cuando la cadena de bloques es muy grande puede llegar a afectar el rendimiento en las aplicaciones. – Como los contratos no se pueden modificar ni borrar, que pasa si alguna condición se modifica, se genera un nuevo contrato pero como se le dice al anterior que ya no aplica. – Es una tecnología que apenas se está incursionando en ella a pesar de que existe hace un tiempo atrás.

Fuente: Propia

Según los pro y contra de cada uno de los modelos definidos descritos anteriormente, se puede concluir que con un solo modelo no se logra mejorar el control de acceso en las aplicaciones desarrolladas a la medida debido a que se tienen casi igual las mismas ventajas y desventajas, sin embargo, la sumatoria de algunas de ellas define un modelo que resuelve la problemática planteada en este proyecto de grado. Los modelos DAC, CapBAC, Blockchain están enfocados para ser utilizados de modo descentralizado por lo que no son compatibles con la propuesta que se

definió en este proyecto y por lo tanto, no fueron incluidos para el cumplimiento del objetivo general.

3.2.3. Estándares de seguridad para la autorización en aplicaciones

Algunos modelos, estándares y protocolos para la implementación de la autenticación y autorización en las aplicaciones desarrolladas a la medida son:

- **SAML Security Assertion Markup Language.**

  Es un estándar abierto basado en Lenguaje de marcado extensible (XML) que se utiliza para intercambiar información de autenticación y autorización de un sistema de identidad a la aplicación de destino basada en la web. Una característica principal es que el token no almacena las credenciales del usuario protegiendo la identidad del usuario [42]. SAML es un estándar basado en XML con el fin de establecer una comunicación entre la autenticación del usuario, los derechos y la información de atributos. El Comité Técnico de Servicios de Seguridad de OASIS es el encargado de definir, mejorar y mantener las especificaciones que se definen en SAML. La versión actual es 2.0 y fue aprobada en marzo de 2005 [54].

  Algunas maneras en las cuales se puede aplicar SAML son [54]:

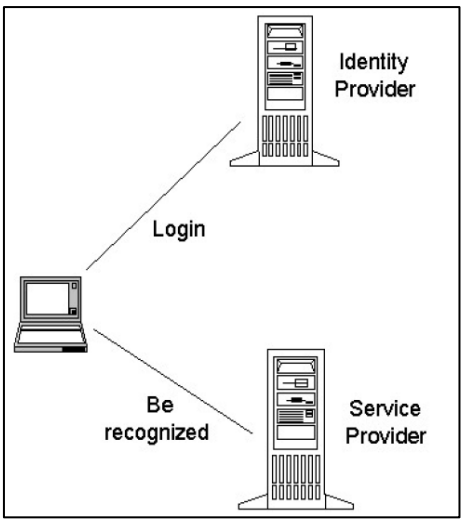
  - ✓ Inicio de sesión único – SSO.
  - ✓ Autorización basada en atributos.
  - ✓ Asegurar servicios web.

  Define 3 roles los cuales se pueden observar en la Figura 3-24 y su respectiva relación entre ellos. El cliente intenta acceder al recurso, el proveedor de servicios solicita al proveedor de identidad si el cliente está autorizado o no para acceder al recurso y finalmente se envía la respuesta al cliente [54]:

  - ✓ Proveedor de servicios: corresponde al recurso que desea acceder el usuario.
  - ✓ Proveedor de identidad: corresponde al servidor de autenticación y autorización.

- ✓ Cliente: aplicación encargada de realizar las solicitudes en nombre del propietario del recurso.

Figura 3-24: Roles SAML.



Fuente: [54]

- **OAUTH**

Protocolo abierto para permitir una autorización segura en un método simple y estándar desde aplicaciones web, móviles y de escritorio. Surgió en noviembre de 2006 debido a que algunos desarrolladores no encontraban un estándar abierto que permitiera la delegación de acceso a una API. Es por ello por lo que se reúnen algunos importantes desarrolladores en esa época (Blaine Cook, Chris Messina, David Recordon, Larry Halff, entre otros). El 3 de octubre de 2007 se lanzó el primer borrador Oauth 1.0. Años más adelante surge la versión 2.0 que corresponde a la versión actual y se está trabajando en la versión 2.1. Uno de los principios que tiene Oauth es que permite dar acceso a ciertos recursos sin compartir la identidad del usuario o información sensible que no es necesaria [55]. El Request for Comments RFC donde se describe y definen los protocolos, métodos, conceptos para el Oauth 2.0 Authorization Framework es el RFC 6749. Oauth no utiliza las credenciales, sino que obtiene un token de acceso donde se detalla alcance, duración y otras propiedades y con este puede acceder a los recursos que necesita en el servidor [56]. Define 4 roles:

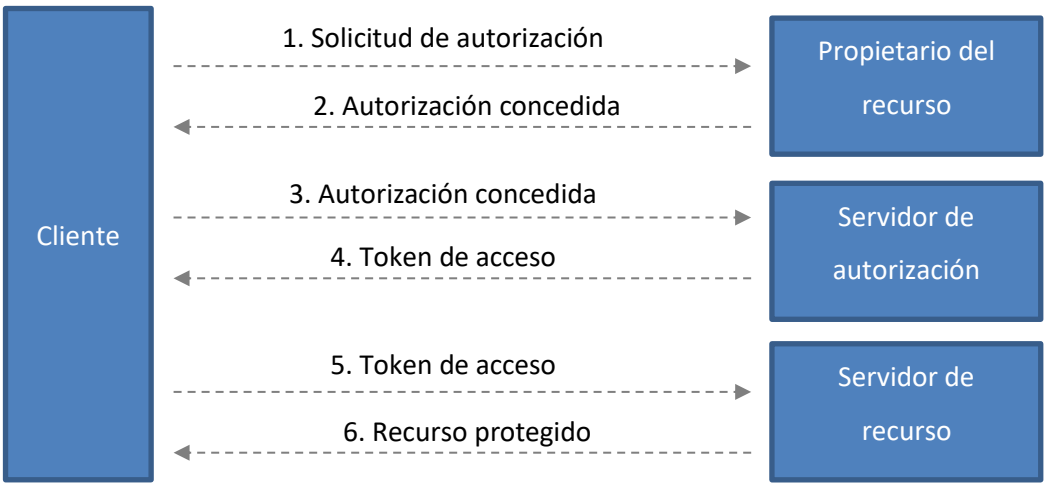
- ✓ Propietario del recurso: Encargado de otorgar el acceso a un recurso protegido.
- ✓ Servidor de recursos: almacena los recursos protegidos.

- ✓ Cliente: aplicación encargada de realizar las solicitudes en nombre del propietario del recurso.
- ✓ Servidor de autorizaciones: servidor encargado de emitir los tokens de acceso al cliente.

A continuación, se describe el flujo del protocolo Oauth, en la Figura 3-25 se puede ver el diagrama.

1. El cliente realiza una solicitud de autorización al propietario del recurso.
2. El propietario del recurso devuelve la autorización al cliente.
3. El cliente envía la autorización entregada por el propietario del recuso al servidor de autorización.
4. El servidor de autorización devuelve el token de acceso al recurso.
5. El token de acceso es enviado al servidor de recurso.
6. El servidor de recurso permite el ingreso del cliente al recurso protegido.

Figura 3-25 Flujo del protocolo Oauth.



Fuente: [56]

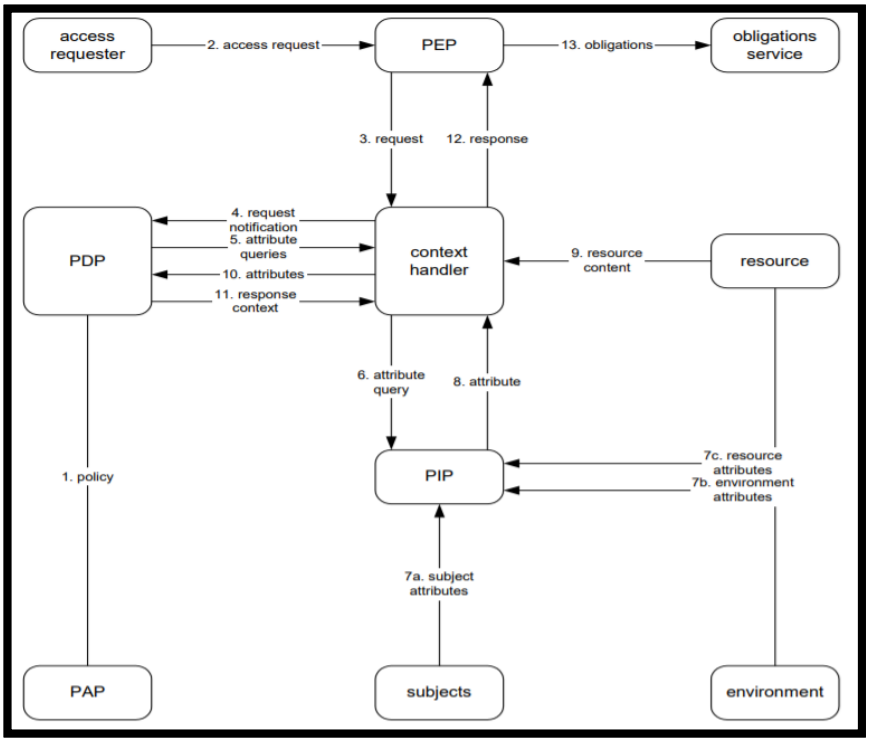
- **XACML**

La sigla XACML significa "Lenguaje de marcado de control de acceso extensible", el equivalente en inglés es "e**X**tensible **A**ccess **C**ontrol **M**arkup **L**anguage". Es un estándar creado por OASIS donde la última versión publicada es la 3.0 en enero de 2013. Describe un lenguaje de políticas y un lenguaje de solicitud / respuesta de decisión de control de acceso, ambas políticas están implementadas en XML. El lenguaje de políticas se utiliza para describir los requisitos generales de control de acceso y tiene puntos de extensión estándar para definir nuevas funciones, tipos de datos, combinar lógica, etc. El lenguaje de solicitud / respuesta le permite formular una consulta para preguntar si se debe permitir una acción determinada e interpretar el resultado. Los modelos ABAC y RBAC pueden ser implementados bajo el estándar XACML [57] .

XACML define las siguientes entidades [44] y su participación en el flujo de datos se puede ver en la Figura 3-26:

- ✓ Context Handler. convierte las solicitudes de decisión en el formato nativo al formato XACML y se comunica con PIP para agregar los atributos requeridos, así mismo, convierte las decisiones de autorización del formato XACML al formato de respuesta.
- ✓ PAP - Policy Administration Point o Punto de Administración de la Política. Es el lugar donde se crean y se administran las políticas de control definidas.
- ✓ PDP - Policy Decision Point o Punto de Decisión de la Política. Es el responsable de validar una solicitud de autorización.
- ✓ PEP - Policy Enforcement Point o Punto de Aplicación de la Política. se encarga del control de acceso generando solicitudes de decisión y aplicando decisiones de autorización, por ejemplo, otorga o deniega el acceso al recurso.
- ✓ PIP - Policy Information Point o Punto de Información de la Política. Esta entidad apoya en la recolección de información adicional en otras fuentes, puede ser por ejemplo: valores de determinados atributos.

Figura 3-26 Diagrama de flujo de datos.



Fuente: [44].

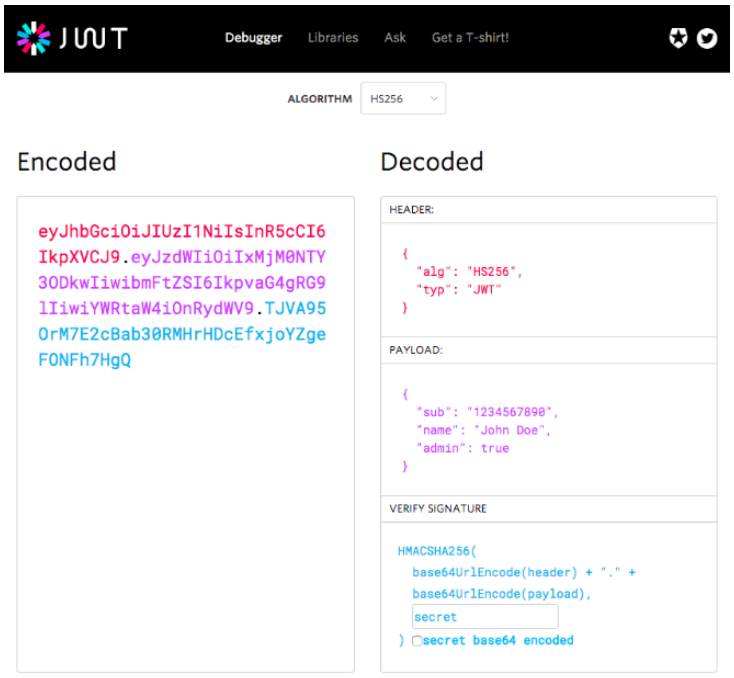
- **JSON/REST for Authorization**

JSON (JavaScript Object Notation) Web Token - JWT, es un formato sencillo y liviano que permite el intercambio de información de manera segura entre dos partes. Fue desarrollado como un estándar abierto. Para tener mayor seguridad de la información transmitida, se puede firmar con una llave secreta o utilizando una clave pública y una clave privada [58], [59].

Un JWT está conformado por tres partes, para ver un ejemplo ir a la Figura 3-27:

- Header
- Payload
- Signature

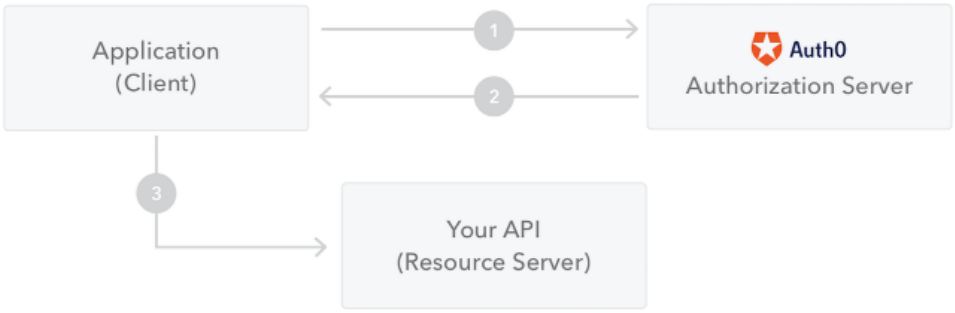
Figura 3-27. Ejemplo de un formato JWT.



Fuente: [58]

En el diagrama que se muestra en la Figura 3-28 se indica cómo se obtiene un JWT y cómo se utiliza para acceder a API o recursos:

Figura 3-28. Diagrama de uso JWT.



Fuente: [58]

1. La aplicación o el cliente solicita autorización al servidor de autorización. Esto se realiza a través de uno de los diferentes flujos de autorización. Por ejemplo, una aplicación web típica compatible con OpenID Connect pasará por el /oauth/authorize punto final utilizando el flujo del código de autorización.
2. Cuando se otorga la autorización, el servidor de autorización devuelve un token de acceso a la aplicación.
3. La aplicación usa el token de acceso para acceder a un recurso protegido (como una API). Fuente: [58]

- **LDAP - Lightweight Directory Access Protocol**

Es un Protocolo Ligero de Acceso a Directorios que tiene como propósito acceder a un servicio de directorios que se ejecuta sobre TCP/IP y está basado en X.500 (estándar de desarrollo para crear directorios producido por la Unión Internacional de Telecomunicaciones, IUT por sus siglas en inglés). Un directorio es un conjunto de objetos con atributos. Se basa en una estructura jerárquica de árbol. La primera versión fue creada en el año 1993 en la Universidad de Michigan, posteriormente surgieron evoluciones, la versión actual es la número 3 creada en el año 2006 y se encuentra definida en el RFC 4511 y la hoja de ruta de especificaciones técnicas en el RFC 4510. [48], [60], [61]. Los servicios que se pueden integrar con el LDAP son: directorios de información, sistemas de autenticación y autorización centralizadas, Active Directory Server de Microsoft, OpenLDAP, Zentyal, entre otros. Así mismo, algunos atributos que se pueden encontrar son: [62]

- uid: identificador del usuario.
- cn: (common name) que corresponde al nombre de la persona o al grupo.
- givenname: es el nombre de pila de la persona.
- sn: (surname) que corresponde al apellido de la persona.
- mail: corresponde a la dirección de correo de la persona.
- O: Organización.
- OU: Unidad organizacional.
- DC: Componente de dominio

Un ejemplo de una estructura de directorios se puede observar en la Figura 3-29, allí se relacionan los atributos mencionados anteriormente:

Figura 3-29. Estructura de directorio LDAP.

Fuente: [48]

3.2.4. Cuadro comparativo de estándares de autorización

De acuerdo con las definiciones de los estándares y protocolos para la autorización de las aplicaciones en el capítulo anterior, se realizó una comparación entre las ventajas y desventajas de cada uno de ellos, esto se puede observar en la Tabla 3-3. No existe un modelo bueno o malo, todo depende de la necesidad y el mecanismo que tengan las aplicaciones al interior de la organización.

Tabla 3-3. Cuadro comparativo de estándares de autorización.

Estándar	Ventajas	Desventajas
SAML	✓ Es un estándar abierto. ✓ Apoya al proceso de autenticación y autorización. ✓ Basado en XML ✓ Enforcado a aplicaciones empresariales. ✓ Autenticación federada.	– No está enfocado a aplicaciones en internet o aplicaciones móviles. – No permite autorización delegada a recursos de Internet
OAUTH	✓ Es un protocolo abierto.	– No se ocupa de la autenticación

Estándar	Ventajas	Desventajas
	✓ Apoya al proceso de autorización. ✓ Permite la delegación de acceso a una API. ✓ Permite dar acceso a ciertos recursos sin compartir la identidad del usuario. ✓ No utiliza las credenciales del usuario. ✓ Enfocado a aplicaciones en internet o aplicaciones móviles. ✓ Basado en JWT.	– No define políticas de control de acceso.
XACML	✓ Es un estándar abierto. ✓ Apoya al proceso de autorización. ✓ Basado en XML ✓ Define políticas de control de acceso. ✓ Las políticas están basadas en los usuarios, recursos, acciones. ✓ Es un lenguaje unificado. ✓ Permite utilizarse en escenarios centralizados o descentralizados.	– No se encarga de la implementación, solo define un diseño. – Puede afectar el rendimiento de la aplicación por las llamadas recurrentes al PDP. – Complejidad de definir y administrar políticas XACML.
JSON	✓ Es un estándar abierto. ✓ Apoya al proceso de autenticación. ✓ Permite el intercambio de información de manera segura entre dos partes.	– No especifica cómo el Cliente obtiene el token. – No tiene servidor de autenticación.

Estándar	Ventajas	Desventajas
	✓ Permite firmar el token con una llave para mayor seguridad. ✓ Formato ligero. ✓ Es un formato de token.	
LDAP	✓ Es un protocolo abierto. ✓ Basado en una estructura jerárquica de árbol. ✓ Apoya al proceso de autenticación de usuarios.	– Solo aplica para uso con directorio activo. – No tiene un proveedor de identidad. – Diseñado para autenticación local. – Los servidores de directorio deben ser compatibles con LDAP.

Fuente: Propia

De acuerdo con lo descrito anteriormente, el estándar LDAP tiene más desventajas que ventajas y el XACML está enfocado más en el diseño que la implementación, por lo cual, estos no fueron tenidos en cuenta en el cumplimiento del objetivo general. Para los protocolos SAML y OAuth, ambos pueden apoyar en la autorización del usuario para acceder al recurso, pero tiene una ventaja mayor OAuth porque utiliza un flujo para la autorización web, es decir, a parte de verificar la identidad solicita un token de acceso, esto entrega otra capa de seguridad entre las aplicaciones, otro punto a favor es que también el protocolo está enfocado a aplicaciones en internet y móviles, lo que ayuda a aquellas organizaciones que se están migrando a la nube.

3.3. Establecimiento de los requerimientos mínimos requeridos para el control de acceso en las aplicaciones a la medida, en cuanto a la autorización.

Con el fin de poder seleccionar el modelo de control de acceso centralizado adecuado para resolver el problema mencionado, se definieron los requerimientos y el detalle del modelo de autorización centralizado basado en prototipos dando cumplimiento al objetivo 3.

3.3.1. Requerimientos de control de acceso

Para la selección de los requisitos que debe cumplir el modelo de control de acceso centralizado se tomaron como referencia:

- ✓ Los resultados de la encuesta y algunas sugerencias de los analistas durante el diligenciamiento de la misma.
- ✓ Validación con algunos expertos sobre el tema, estos fueron tanto arquitectos, analistas como desarrolladores.
- ✓ El estándar de verificación de seguridad de aplicaciones, por sus siglas en inglés ASVS, definido por OWASP [63]
- ✓ La NIST 800.53 revisión 4 [37].

Las necesidades se agruparon en tres funcionalidades con el fin de separar las responsabilidades entre el administrador del componente de autorización como los analistas encargados de administrar la autorización de las aplicaciones de negocio, así mismo, desde la parte de seguridad se incluye un módulo de auditoría o trazabilidad en el uso de la aplicación de autorización centralizada:

- **Administración**: la sección encargada de administrar la aplicación de control de acceso y otorgar el permiso al dueño de la aplicación para que los configure.
- **Autorización**: La sección encargada de configurar el control de acceso a la aplicación de negocio, definir los roles, atributos, niveles de permisos y la segregación de funciones.

- **Auditoría**: La sección se encarga de registrar los eventos realizados en la aplicación de control de acceso, consultar la configuración de autorización de un usuario o una aplicación y generar los reportes para su posterior análisis.

En la Tabla 3-4 se listaron las necesidades o historias de usuario que cumplen con el modelo centralizado de autorizaciones con el propósito de mejorar la seguridad en aplicaciones desarrolladas a la medida, por cada una se asignó su tamaño, su prioridad y se indicó a que funcionalidad hace referencia.

Tabla 3-4. Backlog del producto

ID	Título	Funcionalidades	Tamaño	Prioridad
HU-01	Administrar usuarios a la aplicación de negocio	Autorización	L	Alta
HU-02	Administración de niveles de permisos a la aplicación de negocio	Autorización	M	Alta
HU-06	Administración de roles a la aplicación de negocio	Autorización	L	Alta
HU-07	Administración de atributos a la aplicación de negocio	Autorización	L	Alta
HU-13	Registrar aplicación de negocio para control de acceso centralizado	Administración	L	Alta
HU-14	Parametrización aplicación de control acceso centralizada	Administración	M	Alta
HU-15	Canal de comunicación cifrado	Administración	XS	Alta
HU-19	Integración con sistema de autenticación	Administración	S	Alta
HU-20	Mecanismo de integración con aplicaciones de negocio	Administración	S	Alta
HU-26	Aplicaciones asignadas al responsable de la aplicación de negocio	Autorización	S	Alta

ID	Título	Funcionalidades	Tamaño	Prioridad
HU-03	Fallar de manera segura, los errores no deben generar huecos de seguridad	Administración	M	Media
HU-05	Gestión de Usuario inactivo o retirado	Autorización	L	Media
HU-08	Desaprovisionamiento de usuarios	Administración	M	Media
HU-09	Segregación de funciones	Autorización	XS	Media
HU-16	Manejo de logs	Administración	S	Media
HU-17	Cierre de sesión a la aplicación de negocio	Administración	XS	Media
HU-18	Alta disponibilidad	Administración	S	Media
HU-22	Generación de auditorías	Auditoría	M	Media
HU-04	Creación de roles temporales	Autorización	M	Baja
HU-10	Sistema de control de acceso multifilial	Autorización	S	Baja
HU-11	Copiar roles y niveles de permisos a otra aplicación de negocio	Autorización	S	Baja
HU-12	Replicación de perfiles a usuarios	Autorización	M	Baja
HU-23	Generación de reportes	Auditoría	M	Baja
HU-21	Tiempo de respuesta a la aplicación de negocio	Autorización	S	Baja
HU-24	Respaldo del aplicativo de control de acceso	Administración	XS	Baja
HU-25	Rendimiento del sistema de control de acceso	Administración	S	Baja

Por cada una de las historias de usuario definidas en la tabla anterior, se procedió a detallar cada una con su descripción y criterios de aceptación.

Tabla 3-5. HU-01: Administrar usuarios a la aplicación de negocio

HISTORIA DE USUARIO	
Identificación: HU-01	**Usuario**: Administrador de la aplicación de negocio
Título: Administrar usuarios a la aplicación de negocio	
Prioridad: Alta	**Tamaño**: L
Funcionalidad: Autorización	
Responsable:	
Descripción: **Como:** administrador de la aplicación de negocio, **Quiero:** poder agregar, modificar o retirar los roles o funciones que realizará un usuario en la aplicación , **Para:** que cumplir con las tareas asignadas en su cargo.	
Criterio de aceptación: **CA1**: Agregar usuarios en la aplicación de negocio **Dado**: el formulario de registro de un usuario en una aplicación de negocio, **Cuando**: el administrador de aplicación de negocio vaya a guardar los cambios, **Entonces**: : el sistema debe verificar que el usuario a registrar esté activo y tenga seleccionado una aplicación, haya asignado al menos un rol y en caso de ser varios, que estos roles no pertenezcan a lista de roles incompatibles en la sección de segregación de funciones. El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros. **CA2**: Modificar usuarios en la aplicación de negocio **Dado**: el formulario de edición de un usuario en una aplicación de negocio, **Cuando**: el administrador de aplicación de negocio vaya a guardar los cambios, **Entonces**: el sistema debe verificar que el usuario a registrar está activo, se le han asignado al menos un rol y en caso de ser varios, que estos roles no pertenezcan a lista de roles incompatibles en la sección de segregación de funcione. El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.	

CA3: Eliminar usuarios en la aplicación de negocio
Dado: el formulario de eliminación de un usuario en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a eliminar un usuario,
Entonces: el sistema debe sacar una ventana preguntando si está de acuerdo con eliminar el usuario, el sistema procede a inactivar el usuario en la aplicación asignada. El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA4: Usuario consultado desde el componente de autenticación
Dado: el formulario de registro de un usuario en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a buscar el usuario,
Entonces: el sistema debe retornar la información del componente de autenticación y no que permita crear un usuario local.

CA5: Roles y niveles de permiso
Dado: el formulario de registro de un usuario en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a crear o modificar un usuario,
Entonces: el sistema debe solo mostrar en una lista desplegable los roles y niveles de permisos para esa aplicación de negocio definidos y no los valores asignados a las otras aplicaciones de negocio existentes.

CA6: Consulta de usuarios en la aplicación de negocio
Dado: el formulario de consulta de usuario en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a consultar los usuarios asignados,
Entonces: el sistema debe mostrar los usuarios registrados en esa aplicación de negocio con los roles asignados, estado del usuario y las opciones de editar o eliminar usuario.

Debe estar paginado de a 10 registros por defecto, el usuario puede seleccionar que se pagine por 25 ó 50 o listar todos y poder exportarse la información a Excel.

Tabla 3-6. HU-02: Administración de niveles de permisos a la aplicación de negocio.

HISTORIA DE USUARIO	
Identificación: HU-02	**Usuario**: Administrador de la aplicación de negocio
Título: Administración de niveles de permisos a la aplicación de negocio	
Prioridad: Alta	**Tamaño**: M
Funcionalidad: Autorización	
Responsable:	
Descripción: **Como:** administrador de la aplicación de negocio, **Quiero:** poder crear, modificar o eliminar los niveles de permiso definidos en una aplicación de negocio, **Para:** determinar el tipo de operación que puede realizar el rol sobre el recurso.	
Criterio de aceptación: **CA1**: Agregar niveles de permisos en la aplicación de negocio **Dado**: el formulario de nivel de permisos en una aplicación de negocio, **Cuando**: el administrador de aplicación de negocio vaya a guardar los cambios, **Entonces**: el sistema debe verificar que el usuario haya registrado el nombre del nivel de permiso y la aplicación. El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros. **CA2**: Eliminar niveles de permiso en la aplicación de negocio **Dado**: el formulario de eliminación de niveles de permiso en una aplicación de negocio, **Cuando**: el administrador de aplicación de negocio vaya a eliminar un nivel de permiso, **Entonces**: el sistema debe sacar una ventana preguntando si está de acuerdo con eliminar el nivel de permiso, el sistema procede a validar que ese nivel de permiso no	

esté asignado a algún rol, en caso de encontrarlo, debe informar al usuario que no se puede borrar porque está siendo usado, de lo contrario podrá borrarlo.

CA3: Consulta de niveles de permisos en la aplicación de negocio

Dado: el formulario de consulta de niveles de permisos en una aplicación de negocio,

Cuando: el administrador de aplicación de negocio vaya a consultar los niveles de permisos asignados,

Entonces: el sistema debe mostrar los niveles de permisos registrados en esa aplicación de negocio con los roles asignados, estado del usuario y las opciones de editar o eliminar nivel de permiso.

Debe estar paginado de a 10 registros por defecto, el usuario puede seleccionar que se pagine por 25 ó 50 o listar todos y poder exportarse la información a Excel.

Tabla 3-7. HU-06: Administración de roles a la aplicación de negocio

HISTORIA DE USUARIO	
Identificación: HU-06	**Usuario**: Administrador de la aplicación de negocio
Título: Administración de roles a la aplicación de negocio	
Prioridad: Alta	**Tamaño**: L
Funcionalidad: Autorización	
Responsable:	
Descripción: **Como:** administrador de la aplicación de negocio, **Quiero:** poder crear, modificar o eliminar los roles definidos en una aplicación de negocio, **Para:** determinar el tipo de función que se le puede asignar a un usuario.	
Criterio de aceptación: **CA1**: Agregar roles en la aplicación de negocio **Dado**: el formulario de registro de roles en una aplicación de negocio, **Cuando**: el administrador de aplicación de negocio vaya a guardar los cambios,	

Entonces: el sistema debe verificar que el usuario haya seleccionado la aplicación, el nivel de permiso y diligenciado el nombre del rol a crear.
El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA2: Modificar roles en la aplicación de negocio
Dado: el formulario de edición de un rol en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a guardar los cambios,
Entonces: el sistema debe verificar que el usuario haya seleccionado la aplicación, el nivel de permiso y diligenciado el nombre del rol a modificar.
El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA3: Eliminar roles en la aplicación de negocio
Dado: el formulario de eliminación de un rol en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a eliminar un rol,
Entonces: el sistema debe sacar una ventana preguntando si está de acuerdo con eliminar el rol, el sistema procede a validar que ese rol no esté asignado a algún usuario, en caso de encontrarlo, debe informar al usuario que no se puede borrar porque está siendo usado, de lo contrario podrá borrarlo.
El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA4: Niveles de permiso
Dado: el formulario de registro de un rol en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a crear o modificar un rol,
Entonces: el sistema debe solo mostrar en una lista desplegable los niveles de permisos para esa aplicación de negocio definidos y no las de otras aplicaciones.

CA5: Consulta de roles en la aplicación de negocio
Dado: el formulario de consulta de roles en una aplicación de negocio,

Cuando: el administrador de aplicación de negocio vaya a consultar los roles asignados, **Entonces**: el sistema debe mostrar los roles registrados en esa aplicación de negocio con los roles asignados, con su respectivo nivel de permiso, la aplicación y las opciones de editar o eliminar rol. Debe estar paginado de a 10 registros por defecto, el usuario puede seleccionar que se pagine por 25 ó 50 o listar todos y poder exportarse la información a Excel.

Tabla 3-8. HU.07: Administración de atributos a la aplicación de negocio

HISTORIA DE USUARIO	
Identificación: HU-07	**Usuario**: Administrador de la aplicación de negocio
Título: Administración de atributos a la aplicación de negocio	
Prioridad: Alta	**Tamaño**: L
Funcionalidad: Autorización	
Responsable:	
Descripción: **Como:** administrador de la aplicación de negocio, **Quiero:** poder crear, modificar o eliminar los atributos de usuario o aplicación definidos, **Para:** poder definir las condiciones de uso de la aplicación de negocio.	
Criterio de aceptación: **CA1**: Agregar atributos en la aplicación de negocio **Dado**: el formulario de registro de atributos en una aplicación de negocio, **Cuando**: el administrador de aplicación de negocio vaya a guardar los cambios, **Entonces**: el sistema debe verificar que el usuario haya seleccionado la aplicación, al menos uno de los atributos de área organizacional, cargo, rol, ubicación, dispositivo o tiempo y diligenciado el nombre del atributo. Antes de guardar el atributo, el sistema debe validar si existe alguna configuración similar con los criterios indicados por el usuario para mostrar un mensaje de advertencia indicando las posibles coincidencias y que él decida si quiere guardar la configuración o no.	

El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA2: Modificar atributos en la aplicación de negocio
Dado: el formulario de edición de un atributo en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a guardar los cambios,
Entonces: el sistema debe verificar que el usuario haya seleccionado la aplicación, al menos uno de los atributos de área organizacional, cargo, rol, ubicación, dispositivo o tiempo y diligenciado el nombre del atributo. Antes de guardar el atributo, el sistema debe validar si existe alguna configuración similar con los criterios indicados por el usuario para mostrar un mensaje de advertencia indicando las posibles coincidencias y que él decida si quiere guardar la configuración o no.
El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA3: Eliminar atributos en la aplicación de negocio
Dado: el formulario de eliminación de un atributo en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a eliminar un atributo,
Entonces: el sistema debe sacar una ventana preguntando si está de acuerdo con eliminar el atributo, y si lo estás el sistema procede a borrarlo.
El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA4: Parametrización de atributos
Dado: el formulario de registro de un atributo en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a crear o modificar un atributo,
Entonces: el sistema debe mostrar una lista desplegable con las áreas organizacionales, cargos, roles, ubicación y tipo de dispositivo, y las fechas de inicio y fin en caso de configurar una jornada de uso de la aplicación.

CA5: Consulta de atributos en la aplicación de negocio **Dado**: el formulario de consulta de atributos en una aplicación de negocio, **Cuando**: el administrador de aplicación de negocio vaya a consultar los atributos asignados, **Entonces**: el sistema debe mostrar los atributos registrados en esa aplicación de negocio, la aplicación y las opciones de editar o eliminar atributos. Debe estar paginado de a 10 registros por defecto, el usuario puede seleccionar que se pagine por 25 ó 50 o listar todos y poder exportarse la información a Excel.

Tabla 3-9. HU-13: Registrar aplicación de negocio para control de acceso centralizado.

HISTORIA DE USUARIO	
Identificación: HU-13	**Usuario**: Administrador de la aplicación de control de acceso centralizado
Título: Registrar aplicación de negocio para control de acceso centralizado	
Prioridad: Alta	**Tamaño**: L
Funcionalidad: Administración	
Responsable:	
Descripción: **Como:** administrador de la aplicación de control de acceso centralizado, **Quiero:** poder registrar o eliminar aplicaciones de negocio **Para:** tener la administración centralizada del control de acceso de las aplicaciones desarrolladas a la medida.	
Criterio de aceptación: **CA1**: Crear una aplicación de negocio **Dado**: el formulario de crear una aplicación de negocio, **Cuando**: el administrador de la aplicación de control de acceso centralizado vaya a guardar los cambios, **Entonces**: el sistema debe verificar que los campos de nombre aplicación, la descripción, los responsables (mínimo dos) y la empresa hayan sido diligenciados y	

los responsables estén activos, de lo contrario debe aparecer un mensaje indicando que falta información.
El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA2: Editar aplicación de negocio
Dado: el formulario de editar una aplicación de negocio,
Cuando: el administrador de la aplicación de control de acceso centralizado vaya a guardar los cambios,
Entonces: el sistema debe verificar que los campos de nombre aplicación, la descripción, los responsables (mínimo dos) y la empresa hayan sido diligenciados y los responsables estén activos, de lo contrario debe aparecer un mensaje indicando que falta información.
El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA3: Eliminar administrador de una aplicación de negocio
Dado: el formulario de editar una aplicación de negocio,
Cuando: el administrador de la aplicación de control de acceso centralizado vaya a guardar los cambios,
Entonces: el sistema debe sacar una ventana de confirmación preguntando si está de acuerdo con eliminar el usuario de la aplicación de negocio, el sistema procede a validar si existe otro administrador, en caso de que no, debe informar al usuario que no se puede borrar porque solo hay un administrador asignado, de lo contrario podrá borrarlo. Para guardar los cambios, el sistema debe validar que queden mínimo dos responsables de la aplicación.

CA4: Eliminar aplicación de negocio
Dado: el formulario de editar una aplicación de negocio,
Cuando: el administrador de la aplicación de control de acceso centralizado vaya a eliminar la aplicación,

Entonces: el sistema debe sacar una ventana de confirmación preguntando si está de acuerdo con eliminar la aplicación de negocio y debe escribir la palabra delete para poder borrarla, posterior a esto, el sistema procede a eliminar en toda la base de datos las relaciones asociadas a esa aplicación de negocio, en caso de responder que no, se devuelve al usuario al formulario de editar aplicación. **CA5**: Consulta de aplicaciones de negocio **Dado**: el formulario de consulta de aplicaciones de negocio, **Cuando**: el administrador de la aplicación de control de acceso centralizado vaya a consultar las aplicaciones de negocio, **Entonces**: el sistema debe mostrar las aplicaciones, los responsables y las opciones de editar o eliminar una aplicación de negocio. Debe estar paginado de a 10 registros por defecto, el usuario puede seleccionar que se pagine por 25 ó 50 o listar todos y poder exportarse la información a Excel.

Tabla 3-10. HU-14: Parametrización aplicación de control acceso centralizada

HISTORIA DE USUARIO	
Identificación: HU-14	**Usuario**: Administrador de la aplicación de control de acceso centralizado
Título: Parametrización aplicación de control acceso centralizada	
Prioridad: Alta	**Tamaño**: M
Funcionalidad: Administración	
Responsable:	
Descripción: **Como:** administrador de la aplicación de control de acceso centralizado**,** **Quiero:** poder crear, modificar o eliminar los parámetros de la aplicación de control de acceso centralizada que serán utilizadas por las aplicaciones de negocio, **Para:** que los administradores de aplicación de negocio puedan utilizar las configuraciones de áreas organizacionales, filiales, ubicación, roles, cargo y tipo de dispositivo.	

Criterio de aceptación:

CA1: Agregar parámetros de configuración

Dado: el formulario de agregar parámetros en la aplicación de control de acceso,

Cuando: el administrador de la aplicación de control de acceso centralizado vaya a guardar los cambios,

Entonces: el sistema debe verificar que el usuario haya diligenciado las propiedades de áreas organizacionales, filiales, ubicación, roles, cargo y tipo de dispositivo. Los datos podrán ser diligenciados manualmente o cargados por medio de un archivo plano.

El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA2: Eliminar parámetros de configuración

Dado: el formulario de eliminar parámetros en la aplicación de control de acceso,

Cuando: el administrador de la aplicación de control de acceso centralizado vaya a guardar los cambios,

Entonces: el sistema debe sacar una ventana preguntando si está de acuerdo con eliminar el parámetro de la aplicación de control de acceso, el sistema procede a validar si está siendo usado en algún registro, en caso de que si, debe informar al usuario que no se puede borrar porque está en uso, de lo contrario podrá borrarlo.

CA3: Consulta de parámetros de configuración

Dado: el formulario de consulta de parámetros en la aplicación de control de acceso,

Cuando: el administrador de la aplicación de control de acceso centralizado vaya a consultar los parámetros,

Entonces: el sistema debe mostrar los parámetros de áreas organizacionales, filiales, ubicación, roles, cargo y tipo de dispositivo y las opciones de editar o eliminar.

Debe estar paginado de a 10 registros por defecto, el usuario puede seleccionar que se pagine por 25 ó 50 o listar todos y poder exportarse la información a Excel.

Tabla 3-11. HU-15: Canal de comunicación cifrado

HISTORIA DE USUARIO	
Identificación: HU-15	**Usuario**: Administrador de la aplicación de control de acceso centralizado
Título: Canal de comunicación cifrado	
Prioridad: Alta	**Tamaño**: XS
Funcionalidad: Administración	
Responsable:	
Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** que mis comunicaciones entre el cliente y el servidor sean cifradas, **Para:** evitar robo de identidad o divulgación de información sensible.	
Criterio de aceptación: **CA1**: Comunicación HTTPs y TLS 1.2 **Dado**: el aplicativo de control de acceso, **Cuando**: los usuarios vayan a ingresar a la aplicación, **Entonces**: el sistema debe cargar con certificado, es decir, la dirección web debe iniciar con HTTPs y su comunicación debe ser TLS 1.2.	

Tabla 3-12. HU-19: Integración con sistema de autenticación.

HISTORIA DE USUARIO	
Identificación: HU-19	**Usuario**: Administrador de la aplicación de control de acceso centralizado
Título: Integración con sistema de autenticación	
Prioridad: Alta	**Tamaño**: S
Funcionalidad: Administración	
Responsable:	

Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** que los usuarios sean leídos del directorio activo o desde el sistema de autenticación que tenga la empresa**,** **Para:** garantizar una sola identidad en toda la organización y cumplir con los requisitos de seguridad del componente de autenticación.
Criterio de aceptación: **CA1**: Integración con sistema de autenticación **Dado**: el aplicativo de control de acceso, **Cuando**: los usuarios vayan a ingresar a la aplicación, **Entonces**: el sistema debe integrarse al componente de autenticación de la empresa, es decir, iniciar sesión con su usuario y contraseña de red. Se recomienda habilitar el *Single Sign On* para que el proceso de inicio de sesión sea automático. Si al momento de validar el usuario al sistema de autenticación devuelve que está bloqueado o inactivo no lo puede dejar ingresar en la aplicación y debe mostrarle un mensaje informando esto. **CA2**: Administradores componente centralizado de autorización **Dado**: el aplicativo de control de acceso, **Cuando**: se configuren el administrador de la aplicación de control de acceso centralizado, **Entonces**: el sistema debe solicitar que se registren mínimo dos personas responsables de esta labor para que sea el backup del otro en caso de alguna incapacidad o vacaciones.

Tabla 3-13. HU-20: Mecanismo de integración con aplicaciones de negocio.

HISTORIA DE USUARIO	
Identificación: HU-20	**Usuario**: Administrador de la aplicación de control de acceso centralizado
Título: Mecanismo de integración con aplicaciones de negocio	
Prioridad: Alta	**Tamaño**: S

Funcionalidad: Administración
Responsable:
Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** tener un mecanismo único compatible con cualquier aplicación, **Para:** poder entregar la información en un formato simple y fácil de entender.
Criterio de aceptación: **CA1**: Token de autorización **Dado**: el aplicativo de control de acceso, **Cuando**: la aplicación de negocio consulte los permisos de un usuario, **Entonces**: el sistema debe devolverle a la aplicación de negocio en formato token los roles, niveles de permisos y atributos del usuario configurados en la aplicación de control de acceso centralizado. El mecanismo de negociación debe ser OAuth debido a que algunas de las aplicaciones pueden estar en la nube o ser móviles.

Tabla 3-14. HU-26: Aplicaciones asignadas al responsable de la aplicación de negocio

HISTORIA DE USUARIO	
Identificación: HU-26	**Usuario**: Administrador de la aplicación de negocio
Título: Aplicaciones asignadas al responsable de la aplicación de negocio	
Prioridad: Alta	**Tamaño**: S
Funcionalidad: Administración	
Responsable:	
Descripción: **Como:** aplicación de negocio **Quiero:** que el sistema me muestre las aplicaciones que tengo a cargo **Para:** poder seleccionar alguna y poder administrarla.	

Criterio de aceptación: **CA1**: Visualizar aplicaciones asignadas **Dado**: el aplicativo de control de acceso, **Cuando**: el Administrador de la aplicación de negocio haga clic sobre las aplicaciones a cargo, **Entonces**: el sistema debe mostrarle las aplicaciones de las cuales él es responsable y debe poder tener la opción de seleccionar alguna para cargar los parámetros configurados en las demás opciones de menú en la aplicación de control de acceso. El administrador de la aplicación de negocio solo puede ver la información de su aplicación asignada y no puede ver el módulo de Administración.

Tabla 3-15. HU-03: Fallar de manera segura.

HISTORIA DE USUARIO	
Identificación: HU-03	**Usuario**: Administrador de la aplicación de control de acceso centralizado
Título: Fallar de manera segura, los errores no deben generar huecos de seguridad	
Prioridad: Media	**Tamaño**: M
Funcionalidad: Administración	
Responsable:	
Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** que en caso de presentarse algún error durante la interacción con la aplicación de control de acceso falle de manera segura y correcta, **Para:** evitar que un atacante pueda obtener acceso a la aplicación con privilegios elevados que pueda causar algún incidente de seguridad.	

Criterio de aceptación: **CA1**: Fallar de manera segura **Dado**: el aplicativo de control de acceso, **Cuando**: se presente una falla en la aplicación, **Entonces**: el sistema debe mostrar un mensaje genérico al usuario, no debe mostrar el detalle del error, la sesión del usuario se debe invalidar y liberar todos los recursos abiertos dentro del código como acceso a la base de datos o archivos.

Tabla 3-16. HU-05: Gestión de Usuario inactivo o retirado.

HISTORIA DE USUARIO	
Identificación: HU-05	**Usuario**: Administrador de la aplicación de negocio
Título: Gestión de Usuario inactivo o retirado	
Prioridad: Media	**Tamaño**: L
Funcionalidad: Autorización	
Responsable:	
Descripción: **Como:** administrador de la aplicación de negocio, **Quiero:** que, al intentar otorgar un permiso a un usuario en la aplicación de negocio, si este se encuentra inactivo o retirado no permita asignar ningún rol, **Para:** poder evitar fuga de información innecesariamente.	
Criterio de aceptación: **CA1**: Usuario inactivo durante su creación **Dado**: el formulario de registro de usuarios en una aplicación de negocio, **Cuando**: el administrador de aplicación de negocio vaya a guardar los cambios, **Entonces**: el sistema debe verificar que el usuario al que se le están otorgando los permisos no se lleve a cabo su asignación debido a que se encuentra inactivo por vacaciones, incapacidad o licencia y mostrarle un mensaje al usuario indicando que no se pudo realizar esta acción debido a que el usuario esta inactivo en el sistema de autenticación.	

CA2: Usuario inactivo al momento de consultar la aplicación de negocio

Dado: el aplicativo de negocio,

Cuando: consulte los permisos de un usuario,

Entonces: el sistema debe verificar que el usuario se encuentre activo para poder retornar los permisos asignados de lo contrario, debe devolver un mensaje al usuario indicando que no se pudo realizar esta acción debido a que el usuario esta inactivo en el sistema de autenticación y el aplicativo de control de acceso centralizado, el usuario también debe quedar inactivo.

CA3: Validación del estado del usuario cuando es inactivo

Dado: el aplicativo de negocio,

Cuando: verifique que el usuario se encuentra inactivo por alguna novedad (incapacidad, licencia, vacaciones, entre otros) en el directorio activo,

Entonces: el sistema debe cambiar el estado al usuario de activo a inactivo y a las aplicaciones de negocio que consulten los permisos asignados para ese usuario debe devolver que esta inactivo.

CA4: Validación del estado del usuario cuando es retirado

Dado: el aplicativo de negocio,

Cuando: verifique que el usuario se encuentra retirado en el directorio activo,

Entonces: el sistema debe retirar todos los permisos asignados en todas las aplicaciones que este registrado y poner el estado de activo a inactivo.

CA5: Integración con el Directorio Activo por inactividad del usuario

Dado: el componente de autenticación,

Cuando: un usuario es inactivado por alguna novedad (incapacidad, licencia, vacaciones, entre otros),

Entonces: el componente de autenticación debe consumir un método expuesto por el aplicativo de control de acceso centralizado para notificarle que un usuario se encuentra inactivo y procesa a cambiar el estado al usuario de activo a inactivo y a las

aplicaciones de negocio que consulten los permisos asignados para ese usuario debe devolver que esta inactivo. **CA6**: Integración con el Directorio Activo por activación del usuario **Dado**: el componente de autenticación, **Cuando**: un usuario es puesto activo nuevamente por terminar la novedad (incapacidad, licencia, vacaciones, entre otros), **Entonces**: el componente de autenticación debe consumir un método expuesto por el aplicativo de control de acceso centralizado para notificarle que un usuario se encuentra activo y procesa a cambiar el estado al usuario de inactivo a activo y a las aplicaciones de negocio que consulten los permisos asignados para ese usuario debe devolver los permisos asignados.

Tabla 3-17. HU-08: Desaprovisionamiento de usuarios.

HISTORIA DE USUARIO	
Identificación: HU-08	**Usuario**: Administrador de la aplicación de control de acceso centralizado
Título: Desaprovisionamiento de usuarios	
Prioridad: Media	**Tamaño**: M
Funcionalidad: Administración	
Responsable:	
Descripción: **Como:** aplicación de control de acceso centralizado. **Quiero:** que el sistema de autenticación le avise cuando un usuario es retirado o cambia de área organizacional, **Para:** deshabilitar el usuario o retirarle los permisos asignados con su área anterior y así no permitir que pueda acceder a las aplicaciones que antes tenía a cargo.	

Criterio de aceptación: **CA1**: Retiro de usuario por terminación de contrato **Dado**: el componente de autenticación, **Cuando**: un usuario es retirado del directorio activo porque su contrato fue terminado, **Entonces**: el componente de autenticación debe consumir un método expuesto por el aplicativo de control de acceso centralizado para notificarle que un usuario ha sido de baja y procesa a retirarle todos los permisos asignados en las aplicaciones de negocio y colocarle el estado de inactivo. **Nota:** Para lograr el correcto desaprovisionamiento del usuario, la aplicación de nómina debe notificar la novedad de retiro de manera oportuna para que el componente de control de acceso centralizado pueda realizar los cambios respectivos.

Tabla 3-18. HU-09: Segregación de funciones.

HISTORIA DE USUARIO	
Identificación: HU-09	**Usuario**: Administrador de la aplicación de negocio
Título: Segregación de funciones	
Prioridad: Media	**Tamaño**: XS
Funcionalidad: Autorización	
Responsable:	
Descripción: **Como:** administrador de la aplicación de negocio, **Quiero:** que, al intentar otorgar un permiso a un usuario en la aplicación de negocio, si está en la lista de matriz de incompatibilidad de roles no puede ser asignado al usuario, **Para:** poder evitar fuga de información innecesariamente.	

Criterio de aceptación: **CA1**: Matriz de incompatibilidad de roles **Dado**: el formulario de segregación de funciones en la aplicación de control de acceso, **Cuando**: el administrador de la aplicación de negocio vaya a guardar los cambios, **Entonces**: el sistema debe validar que haya seleccionado los dos roles respectivos incompatibles. Si ambos roles son iguales, el sistema debe notificarle al usuario que no se puede guardar porque no hay asignado un rol diferente de el mismo para guardarse. El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros. **CA2**: Consulta de Roles **Dado**: el formulario de segregación de funciones en la aplicación de control de acceso, **Cuando**: el administrador de aplicación de negocio vaya a crear o modificar un usuario, **Entonces**: el sistema debe solo mostrar en las listas desplegables los roles configurados para esa aplicación de negocio definidos y no las de otras aplicaciones.

Tabla 3-19. HU-16: Manejo de logs.

HISTORIA DE USUARIO	
Identificación: HU-16	**Usuario**: Administrador de la aplicación de control de acceso centralizado
Título: Manejo de logs	
Prioridad: Media	**Tamaño**: S
Funcionalidad: Administración	
Responsable:	
Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** que el sistema tenga un log de aplicación, **Para:** identificar en caso de un fallo en el sistema que está ocurriendo y poder dar con la solución.	

Criterio de aceptación: **CA1**: Generación de log en la aplicación de control de acceso centralizado **Dado**: el aplicativo de control de acceso, **Cuando**: se presente una falla en la aplicación, **Entonces**: el sistema debe mostrar un mensaje genérico al usuario, el detalle del error debe ser almacenado en un archivo plano y liberar todos los recursos abiertos dentro del código como acceso a la base de datos o archivos. No se debe guardar información sensible como contraseñas, hash o llaves API. Se debe manejar las excepciones por la estructura: *try, catch y finally*. **CA2**: Acceso al log de la aplicación de control de acceso centralizado **Dado**: el aplicativo de control de acceso, **Cuando**: se presente una falla en la aplicación, **Entonces**: el log generado por la aplicación solo debe ser visible por los administradores de la aplicación de control de acceso. **CA3**: Estructura del log en la aplicación de control de acceso centralizado **Dado**: el aplicativo de control de acceso, **Cuando**: se presente una falla en la aplicación, **Entonces**: el sistema debe registrar la siguiente información en el archivo log: Fecha, el error completo arrojado por la aplicación, el nivel de severidad. El nombre del archivo debe ser AAAMMDD_Log.txt Cada día debe generar su propio archivo de log.

Tabla 3-20. HU-17: Cierre de sesión a la aplicación de negocio.

HISTORIA DE USUARIO	
Identificación: HU-17	**Usuario**: Administrador de la aplicación de control de acceso centralizado
Título: Cierre de sesión a la aplicación de negocio	
Prioridad: Media	**Tamaño**: XS
Funcionalidad: Administración	

Responsable:
Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** que el sistema tenga una opción para cierre de sesión o que se cierre la sesión por inactividad, **Para:** evitar que un atacante pueda ingresar sin autorización y realizar alguna modificación causando un incidente de seguridad.
Criterio de aceptación: **CA1**: Cierre de sesión por el usuario **Dado**: el aplicativo de control de acceso, **Cuando**: el usuario haga clic en cerrar sesión, **Entonces**: el sistema debe cerrar la sesión e invalidar la sesión. Para poder ingresar nuevamente debe digitar su usuario y contraseña. Debe mostrarle al usuario un mensaje indicando que el cierre de sesión fue exitoso. **CA2**: Cierre de sesión por inactividad **Dado**: el aplicativo de control de acceso, **Cuando**: detecte que el usuario lleva más de 5 minutos sin cambiar de página, **Entonces**: el sistema debe notificarle al usuario si desea continuar en el sitio, sino recibe respuesta en un minuto, debe cerrar la sesión e invalidar la sesión. El tiempo de validez de la sesión debe ser configurable y poderse cambiar en caso de ser requerido.

Tabla 3-21. HU-18: Alta disponibilidad.

HISTORIA DE USUARIO	
Identificación: HU-18	**Usuario**: Administrador de la aplicación de control de acceso centralizado
Título: Alta disponibilidad	
Prioridad: Media	**Tamaño**: S
Funcionalidad: Administración	

Responsable:
Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** que el sistema este en alta disponibilidad, **Para:** evitar que las aplicaciones de negocio puedan tener una indisponibilidad porque la aplicación de control de acceso centralizada no es accesible.
Criterio de aceptación: **CA1**: Alta disponibilidad **Dado**: el aplicativo de control de acceso, **Cuando**: uno de los servidores no se encuentre operando, **Entonces**: el sistema debe subir el nodo inactivo para que pueda continuar con la prestación del servicio.

Tabla 3-22. HU-22: Generación de auditorías.

HISTORIA DE USUARIO	
Identificación: HU-22	**Usuario**: Administrador de la aplicación de control de acceso centralizado o Administrador de la aplicación de negocio
Título: Generación de auditorías	
Prioridad: Media	**Tamaño**: M
Funcionalidad: Auditoría	
Responsable:	
Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** que el sistema maneje la bitácora de trazabilidad y seguimiento a las acciones realizadas en la aplicación de control de acceso, **Para:** poder detectar algún comportamiento anómalo en línea por medio de un SOC o dar respuesta a algún incidente de seguridad.	
Criterio de aceptación: **CA1**: Generación de Auditorías **Dado**: el aplicativo de control de acceso,	

Cuando: se realice alguna acción de ingreso, modificación o eliminación de un rol, nivel de permiso, atributos, usuario, segregación de funciones, parámetros de la aplicación, **Entonces**: el sistema debe guardar en la base de datos estas acciones indicando quién hizo el cambio, descripción del cambio, la funcionalidad afectada, la acción realizada y en qué fecha. **CA2**: Consulta de Auditorías **Dado**: el aplicativo de control de acceso, **Cuando**: uno de los usuarios genere consulta de auditoría, **Entonces**: el sistema debe mostrar la información solamente de las aplicaciones que tiene a cargo. El Administrador de la aplicación de control de acceso centralizado puede generar cualquier reporte independiente de la aplicación. **CA3**: Integración con el SOC **Dado**: el aplicativo de control de acceso, **Cuando**: se registre algún evento de auditoría **Entonces**: el sistema reportarle al SOC dicho ajuste para que pueda identificar si existe algún comportamiento atípico de un usuario.

Tabla 3-23. HU-04: Creación de roles temporales.

HISTORIA DE USUARIO	
Identificación: HU-04	**Usuario**: Administrador de la aplicación de negocio
Título: Creación de roles temporales	
Prioridad: Baja	**Tamaño**: M
Funcionalidad: Autorización	
Responsable:	

Descripción:

Como: aplicación de control de acceso centralizado,

Quiero: que la aplicación de control de acceso permita asignar roles por un período de tiempo,

Para: optimizar la delegación de permisos de un usuario a otro por unas vacaciones o incapacidad de forma rápida.

Criterio de aceptación:

CA1: Agregar usuario por un período de tiempo

Dado: el formulario de registro de un usuario temporal en una aplicación de negocio,

Cuando: el administrador de aplicación de negocio vaya a guardar los cambios,

Entonces: el sistema debe verificar que el usuario a registrar este activo, haya seleccionado una aplicación, haya asignado al menos un rol y en caso de ser varios, que estos roles no pertenezcan a lista de roles incompatibles en la sección de segregación de funciones y haya indicado el tiempo de vigencia de ese permiso en la aplicación. Las fechas de inicio y fin no deben ser inferior a la actual.

El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA2: Modificar usuario por un período de tiempo

Dado: el formulario de edición de un usuario temporal en una aplicación de negocio,

Cuando: el administrador de aplicación de negocio vaya a guardar los cambios,

Entonces: el sistema debe verificar que el usuario a registrar este activo, haya seleccionado una aplicación, haya asignado al menos un rol y en caso de ser varios, que estos roles no pertenezcan a lista de roles incompatibles en la sección de segregación de funciones y haya indicado el tiempo de vigencia de ese permiso en la aplicación. Las fechas de inicio y fin no deben ser inferior a la actual.

El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA3: Eliminar usuario por un período de tiempo

Dado: el formulario de eliminación de un usuario temporal en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a eliminar un usuario,
Entonces: el sistema debe sacar una ventana preguntando si está de acuerdo con eliminar el usuario, el sistema procede a inactivar el usuario en la aplicación asignada. Si el usuario intenta borrar el registro una vez pasada la fecha actual, el sistema no lo debe dejar y mostrarle un mensaje de que no fue posible borrarlo.
El sistema debe validar los datos de entrada tanto del lado del servidor como del cliente para evitar vulnerabilidades como SQL Injection, XSS, entre otros.

CA4: Usuario consultado desde el componente de autenticación
Dado: el formulario de registro de un usuario en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a buscar el usuario,
Entonces: el sistema debe retornar la información del componente de autenticación y no que permita crear un usuario local.

CA5: Roles y niveles de permiso
Dado: el formulario de registro de un usuario en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a crear o modificar un usuario,
Entonces: el sistema debe solo mostrar en una lista desplegable los roles y niveles de permisos para esa aplicación de negocio definidos y no las de otras aplicaciones.

CA6: Consulta de usuarios temporales en la aplicación de negocio
Dado: el formulario de consulta de usuario temporal en una aplicación de negocio,
Cuando: el administrador de aplicación de negocio vaya a consultar los usuarios temporales asignados,
Entonces: el sistema debe mostrar los usuarios temporalmente registrados en esa aplicación de negocio con los roles asignados, la fecha de inicio, la fecha de fin, y las opciones de editar o eliminar usuario.

Debe estar paginado de a 10 registros por defecto, el usuario puede seleccionar que se pagine por 25 ó 50 o listar todos y poder exportarse la información a Excel.

Tabla 3-24. HU-10: Sistema de control de acceso multifilial.

HISTORIA DE USUARIO	
Identificación: HU-10	**Usuario**: Administrador de la aplicación de negocio
Título: Sistema de control de acceso multifilial	
Prioridad: Baja	**Tamaño**: S
Funcionalidad: Autorización	
Responsable:	
Descripción: **Como:** aplicación de negocio, **Quiero:** que la aplicación permita configurar los mismos permisos en empresas diferentes, **Para:** permitir utilizar el mismo modelo, solo cambiando los parámetros de configuración.	
Criterio de aceptación: **CA1**: Aplicación Multifilial **Dado**: el aplicativo de control de acceso, **Cuando**: se requiera configurar los mismos permisos para otra filial o empresa, **Entonces**: el sistema debe permitir seleccionar esa configuración a que empresa le pertenece.	

Tabla 3-25. HU-11: Copiar roles y niveles de permisos a otra aplicación de negocio.

HISTORIA DE USUARIO	
Identificación: HU-11	**Usuario**: Administrador de la aplicación de negocio
Título: Copiar roles y niveles de permisos a otra aplicación de negocio	
Prioridad: Baja	**Tamaño**: S
Funcionalidad: Autorización	
Responsable:	

Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** que la aplicación de control de acceso permita crear los mismos roles y niveles de permisos, **Para:** ser asignados a una nueva aplicación de negocio, sin necesidad de tener que configurar manualmente esto.
Criterio de aceptación: **CA1**: Replicación de permisos en otra aplicación **Dado**: el aplicativo de control de acceso centralizado, **Cuando**: se requiera configurar los mismos permisos para otra aplicación, **Entonces**: el sistema debe permitir seleccionar esa configuración y replicarla a la otra aplicación.

Tabla 3-26. HU-12: Replicación de perfiles a usuarios.

HISTORIA DE USUARIO	
Identificación: HU-12	**Usuario**: Administrador de la aplicación de negocio
Título: Replicación de perfiles a usuarios	
Prioridad: Baja	**Tamaño**: M
Funcionalidad: Autorización	
Responsable:	
Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** que la aplicación de control de acceso permita asignar los mismos roles de un usuario a otro, **Para:** optimizar la asignación de permisos a un usuario que va a realizar las mismas funciones.	
Criterio de aceptación: **CA1**: Replicación de permisos a otro usuario **Dado**: el aplicativo de control de acceso centralizado, **Cuando**: se requiera configurar los mismos roles a otro usuario,	

Entonces: el sistema debe permitir copiar los roles que tiene un usuario asignado a otro. El sistema debe disponer de un buscador para seleccionar el usuario y la aplicación a la cual se le desean copiar los permisos para ser asignados a otro usuario. **CA2**: Validación de la segregación de funciones **Dado**: el aplicativo de control de acceso centralizado, **Cuando**: se requiera configurar los mismos roles a otro usuario, **Entonces**: el sistema debe validar que roles tiene asignados actualmente para esa aplicación, en caso de que existe alguna incompatibilidad entre los permisos actuales y los nuevos que se van a asignar, el sistema debe mostrarle al usuario los roles incompatibles para que el usuario seleccione cual rol debe dejarle al usuario y poder completar la asignación, en caso de que no seleccione el rol a eliminar el proceso no se debe hacer.

Tabla 3-27. HU-23: Generación de reportes.

HISTORIA DE USUARIO	
Identificación: HU-23	**Usuario**: Administrador de la aplicación de control de acceso centralizado o Administrador de la aplicación de negocio
Título: Generación de reportes	
Prioridad: Baja	**Tamaño**: M
Funcionalidad: Auditoría	
Responsable:	
Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** que la aplicación permita generar reportes, **Para:** identificar los usuarios registrados en la aplicación con sus respectivos roles y niveles de permisos.	

Criterio de aceptación: **CA1**: Consulta de reportes **Dado**: el aplicativo de control de acceso, **Cuando**: uno de los usuarios genere consulta sobre los permisos asignados en una aplicación, **Entonces**: el sistema debe mostrarle la información solamente de las aplicaciones que tiene a cargo. El Administrador de la aplicación de control de acceso centralizado puede generar cualquier reporte independiente de la aplicación.

Tabla 3-28. HU-21: Tiempo de respuesta.

HISTORIA DE USUARIO	
Identificación: HU-21	**Usuario**: Administrador de la aplicación de negocio
Título: Tiempo de respuesta a la aplicación de negocio	
Prioridad: Baja	**Tamaño**: S
Funcionalidad: Autorización	
Responsable:	
Descripción: **Como:** aplicación de negocio, **Quiero:** que el sistema entregue a las aplicaciones la información en un tiempo menor a 5 segundos, **Para:** evitar lentitud en las aplicaciones de negocio.	
Criterio de aceptación: **CA1**: Tiempo de respuesta a la aplicación de negocio **Dado**: el aplicativo de negocio, **Cuando**: consulte sobre los permisos asignados a un usuario en específico, **Entonces**: el sistema debe retornarle la información en un tiempo menor a 5 segundos.	

Tabla 3-29. HU-24: Respaldo del sistema de control de acceso.

HISTORIA DE USUARIO	
Identificación: HU-24	**Usuario**: Administrador de la aplicación de control de acceso centralizado
Título: Respaldo del aplicativo de control de acceso	
Prioridad: Baja	**Tamaño**: XS
Funcionalidad: Administración	
Responsable:	
Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** que el sistema sea respaldado todos los días, **Para:** que en caso de falla y se deba restaurar la información desde un respaldo este sea lo más reciente posible para evitar pérdida de información.	
Criterio de aceptación: **CA1**: Respaldo a la aplicación de control de acceso centralizado **Dado**: el aplicativo de control de acceso centralizado, **Cuando**: se presente una falla en los datos y no se pueda restablecer el servicio, **Entonces**: el sistema debe restaurarse desde un respaldo del día anterior	

Tabla 3-30. HU-25: Rendimiento de control de acceso**.**

HISTORIA DE USUARIO	
Identificación: HU-25	**Usuario**: Administrador de la aplicación de control de acceso centralizado
Título: Rendimiento de control de acceso	
Prioridad: Baja	**Tamaño**: S
Funcionalidad: Administración	
Responsable:	
Descripción: **Como:** aplicación de control de acceso centralizado, **Quiero:** que el sistema soporte más de 2.000 solicitudes concurrentes,	

Para: que no se genere indisponibilidad en las aplicaciones de negocio cuando consulten los permisos de un usuario.
Criterio de aceptación: **CA1**: Rendimiento de la aplicación **Dado**: el aplicativo de control de acceso centralizado, **Cuando**: se realicen máximo 2.000 peticiones al tiempo, **Entonces**: el sistema debe poder entregar la información solicitada a las aplicaciones de negocio.

3.3.2. Modelo propuesto para la centralización de la autorización en las aplicaciones

Los usuarios ejecutan funciones que van de acuerdo con su cargo y que se pueden modelar en los sistemas por medio del RBAC, sin embargo, algunas pueden ser algo amplias lo que permite dejar puertas abiertas en las aplicaciones y un atacante llegar a aprovecharse de esto. Es por ello que, para acotarlo se procede a combinar varios modelos como son ABAC y el CBAC, porque se logra por medio de los atributos como cargo, área, hora, dirección ip, ubicación, dispositivo electrónico utilizado, usuario activo, entre otros cerrar o disminuir las brechas de seguridad como el caso de una persona malintencionada que quiera acceder a la información con el propósito de modificarla, eliminarla o descargarla y así obtener algún beneficio económico. Los modelos como DAC, CapBAC, Blockchain se enfoncan en otro esquema diferente a las aplicaciones desarrolladas a la medida, por ejemplo, son descentralizados, enfocados a recursos como un intercambio de archivos, para el caso de CapBAC tiene como desventaja la dificultad para conocer quién puede acceder a un recurso y revocar todos los permisos. El modelo PBAC está enfocado en las rutas de la aplicaciones y algunas pueden tener muchas de ellas generando complejidad en su administración y control. Y por último, este modelo no está orientado a la tecnología Blockchain.

Con base en lo encontrado en el capítulo 3.2, donde se presentaron los diferentes modelos de control de acceso y se realizó un cuadro comparativo entre ellos con sus ventajas y desventajas, se concluye que un único modelo no es suficiente porque no se satisfacen todos los requisitos definidos; se requiere en general un modelo que permita el control de acceso tanto de manera estática, basado en políticas prestablecidas, como dinámicas, basado en el comportamiento y contexto del usuario y que se integre con el sistema de autenticacion encargado de gestionar los usuarios para que pueda saberse si el usuario esta activo, inactivo o retirado de la organización. Como resultado de lo anterior, se concluyó que la mejor práctica para definir el modelo centralizado de autorización es usar los modelos RBAC, ABAC y CBAC.

Al disponer de un sólo modelo centralizado de autorizaciones en la organización se permite tener el control en la asignación o retiro de los permisos, identificando los usuarios a que aplicaciones están accediendo y con qué nivel de permisos; así mismo, al integrarse con el componente de

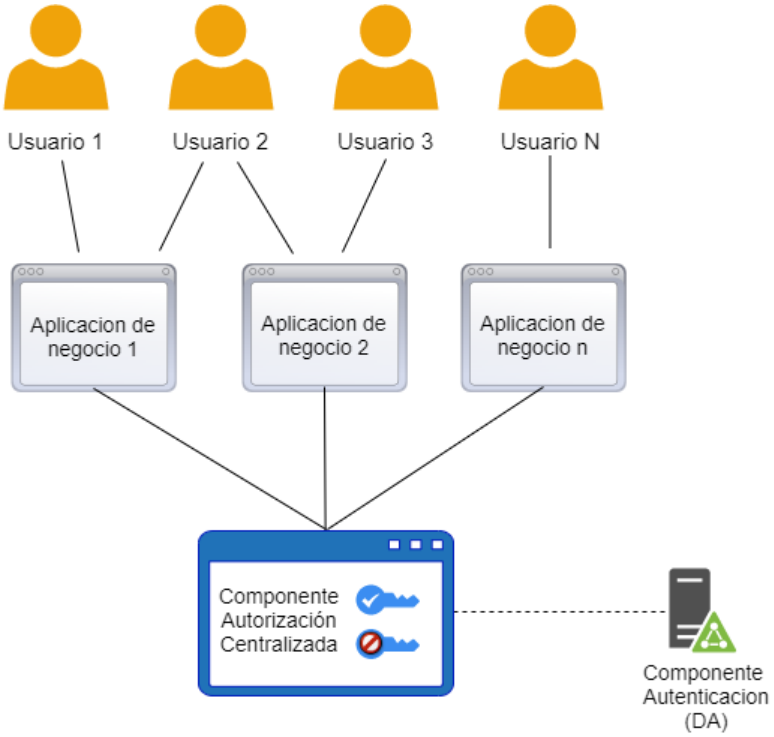
autenticación, se valida que el usuario este activo para poder que continue con sus funciones en las aplicaciones de negocio, de lo contrario el modelo va a retornar que no está autorizado para accederla, y finalmente, al estar centralizado, se puede hacer una auditoría y monitoreo a los usuarios tanto en la asignación, modificación como retiro de sus permisos, quién lo llevo a cabo y cuándo fue realizado.

El modelo centralizado a alto nivel se puede observar en la Figura 3-30, en la cual los usuarios se conectan a las aplicaciones de negocio, ellas a su vez llaman por medio del consumo de APIs al componente de autorización centralizado para que les retorne los permisos que tienen los usuarios registrados para esas aplicaciones de negocio.

El componente de autorización centralizado tiene dos funciones principales: validar la autenticación e identidad del usuario y retornar los permisos asignados a las aplicaciones de negocios. El proceso inicia por preguntarle al componente de autenticación si la información enviada del usuario es correcta, si existe y que se encuentre activo; si todo es correcto el componente procede a ir al modelo centralizado ARC-BAC con el fin de determinar los permisos que tiene el usuario para esa aplicación respectiva, de lo contrario devuelve un mensaje de advertencia indicando que el usuario está inactivo y no puede acceder a la aplicación. Como prerrequisito para que pueda funcionar el modelo, la aplicación de negocio debe haber registrado previamente los roles y atributos definidos para el manejo de su autorización. Cuando la aplicación de negocio llame al componente de autorización para que le retorne la información de los permisos configurados para un usuario, su mecanismo de comunicación debe ser a través del protocolo Oauth debido a que las aplicaciones pueden estar en internet o incluso ser aplicaciones móviles, y la respuesta con la información entregada del componente de autorización a la aplicación de negocio debe ser en un formato simple como json.

Este modelo es dependiente del componente de Autenticación definido por la organización, el cual es el repositorio único de autenticación de los usuarios en la red corporativa. El modelo de autorización no considera que tenga su propia base de datos de usuarios.

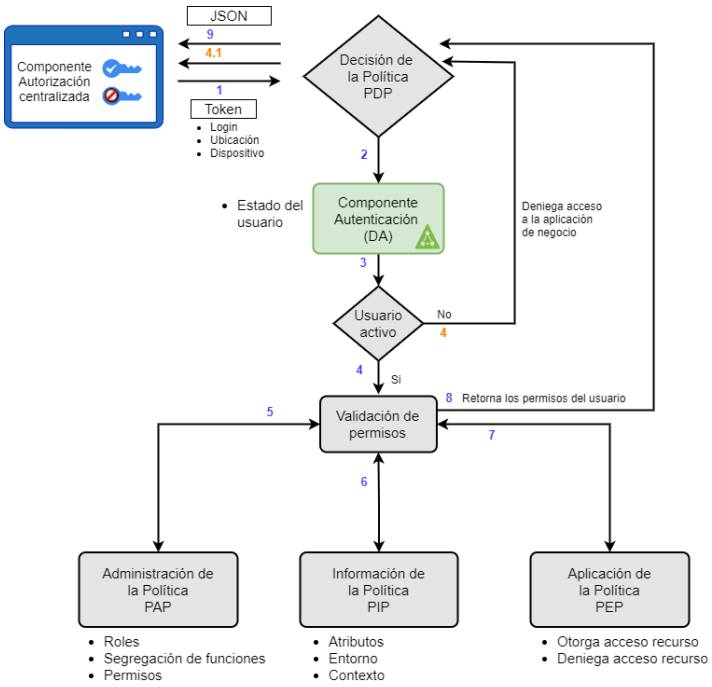
Figura 3-30. Modelo centralizado de autorizaciones a alto nivel.



Fuente: propia.

Para diseñar los procesos que participan en el Componente de Autorización Centralizada se tomó como referencia el estándar XACML y se ajustó a las necesidades del proyecto como se puede observar en la Figura 3-31. El modelo RBAC se aplica en el proceso PAP, allí se define el nivel de permisos, los roles y la segregación de funciones. Los modelos ABAC y CBAC se encuentran dentro del proceso PIP y se parametrizan los atributos, las variables de entorno y contexto que se van a utilizar en el manejo de la autorización de la aplicación de negocio.

Figura 3-31. Modelo centralizado de autorizaciones para mejorar la seguridad en aplicaciones desarrolladas a la medida en las organizaciones.



Fuente: Propia.

A continuación, se detallan los procesos involucrados en el modelo centralizado de autorización en las aplicaciones propuesto:

- PDP - Punto de Decisión de la Política. Es el proceso encargado de recibir la solicitud de autorización de un usuario en la aplicación y el orquestador para lanzar la llamada a los demás procesos como lo son Componente de Autenticación (DA), PAP, PEP, PIP y con base a los resultados arrojados por estos determina la viabilidad o no de acceder al recurso solicitado y envía la respuesta a la aplicación.

- Componente de Autenticación: Es el proceso encargado de validar la autenticación, identidad y el estado del usuario que solicita el acceso al recurso; corresponde a la integración con el sistema autenticador de la empresa para conocer si el usuario está activo, inactivo o retirado de la organización. En caso de que el usuario esté retirado, se ejecuta el proceso de inactivación en los diferentes aplicativos que tenga permisos de

acceso y envía al PDP su resultado. Si está activo, continua con el proceso de validación de permisos.

- Validación de permisos: Encargado de llamar secuencialmente al PAP, PIP y PEP para construir los permisos del usuario y devolverlo al PDP.

- PAP – Punto de Aplicación de la Política. Es el proceso encargado de realizar la validación de manera estática, basado en los roles del usuario, niveles de permisos y la segregación de funciones, para posteriormente enviar la respuesta al proceso intermedio validación de permisos.

- PIP - Punto de Información de la Política. Es el proceso encargado de realizar la validación en tiempo de ejecución de la solicitud por los atributos definidos como ubicación, tipo dispositivo, fecha y hora de la solicitud, para posteriormente enviar la respuesta a la validación de permisos si el usuario está autorizando o no.

- PEP - Punto de Aplicación de la Política. Es el proceso encargado de aplicar la política sobre el recurso solicitado por la aplicación, determinando si el usuario puede acceder o no al recurso para enviar posteriormente la respuesta al proceso intermedio validación de permisos.

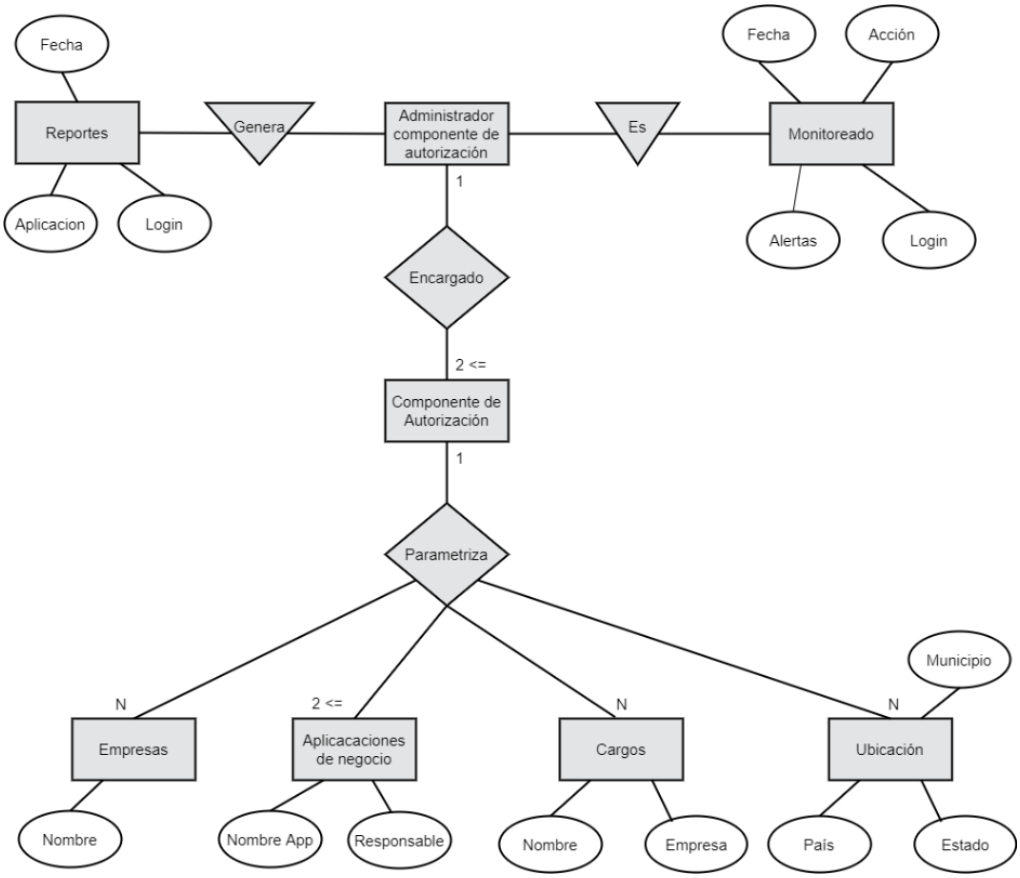
En la Figura 3-32 se encuentra el diagrama entidad relación entre el administrador de la aplicación de control de acceso centralizado y las entidades con las que se relaciona con el fin de obtener una perspectiva general de la operación del responsable del componente de control de acceso. El administrador componente de autorización debe tener mínimo dos responsables y se relaciona con tres (3) entidades:

- Componente de Autorización: encargado de parametrizar la aplicación, allí se registran las aplicaciones de negocio (debe existir mínimo dos responsables de ésta), la (s) empresa(s), cargos organizacionales y la ubicación de la(s) empresa(s).

- Monitoreado: las acciones realizadas por el administrador de la aplicación de control de acceso centralizado son auditados y se guardar en un archivo para un análisis posterior en caso de ser requerido, los datos mínimos almacenados son Fecha, Login y Acción. Dentro de estas acciones si se detecta algún comportamiento anómalo se puede generar alertas a un grupo o usuarios que se indiquen. Así mismo, se puede llevar a un SOC *Security Operation Center* para ser monitoreado y también se generen alertas tempranas de alguna situación atípica.

- Reportes: permite realizar consultas sobre las acciones realizadas por el administrador de la aplicación de negocio o reportes sobre una aplicación con sus respectivos permisos. Los datos mínimos de consulta son Fecha, Aplicación y Login.

Figura 3-32. Modelo entidad relación del administrador aplicación control de acceso.



Fuente: Propia.

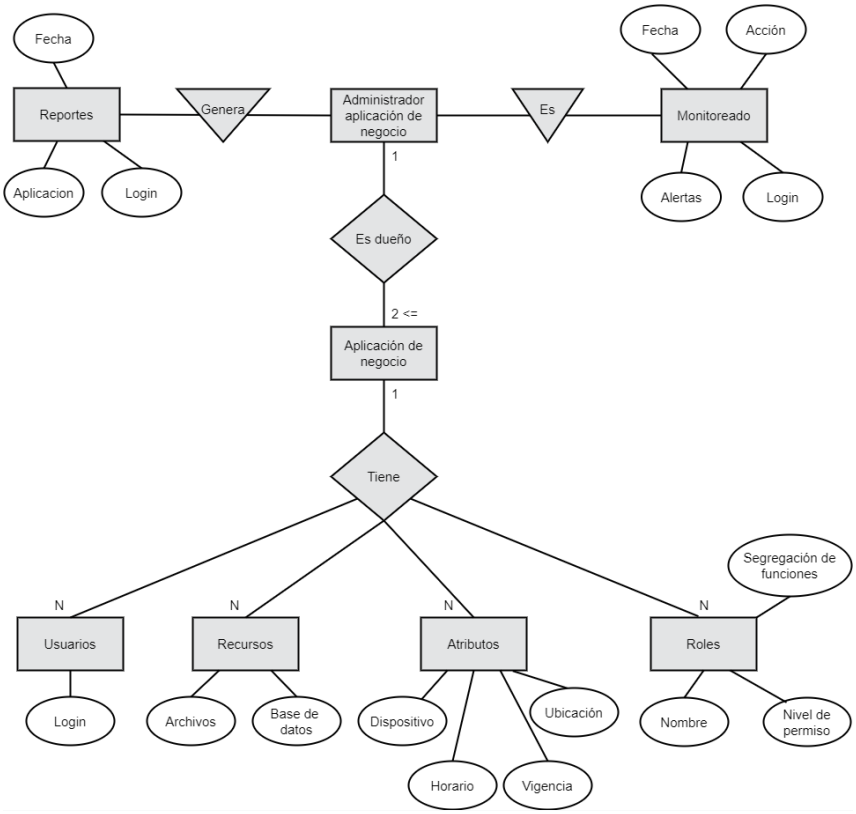
En la Figura 3-33 se encuentra el diagrama entidad relación entre el administrador de la aplicación de negocio y las entidades con las que se relaciona. El administrador de la aplicación del negocio es el encargado de configurar el control de acceso para la aplicación de negocio, debe tener mínimo dos responsables y se relaciona con tres (3) entidades:

- Aplicación de negocio: el responsable de la aplicación de negocio administra los roles, recursos, atributos que tiene la aplicación y usuarios que interactúan con este.

- Monitoreado: las acciones realizadas por el administrador de la aplicación de control de acceso centralizado son auditados y se guardar en un archivo para un análisis posterior en caso de ser requerido, los datos mínimos almacenados son Fecha, Login y Acción. Dentro de estas acciones si se detecta algún comportamiento anómalo se puede generar alertas a un grupo o usuarios que se indiquen. Así mismo, se puede llevar a un SOC *Security Operation Center* para ser monitoreado y también se generen alertas tempranas de alguna situación atípica.

- Reportes: permite realizar consultas sobre las acciones realizadas por el administrador de la aplicación de negocio o reportes sobre una aplicación con sus respectivos permisos. Los datos mínimos de consulta son Fecha, Aplicación y Login.

Figura 3-33. Modelo entidad relación del administrador aplicación de negocio.



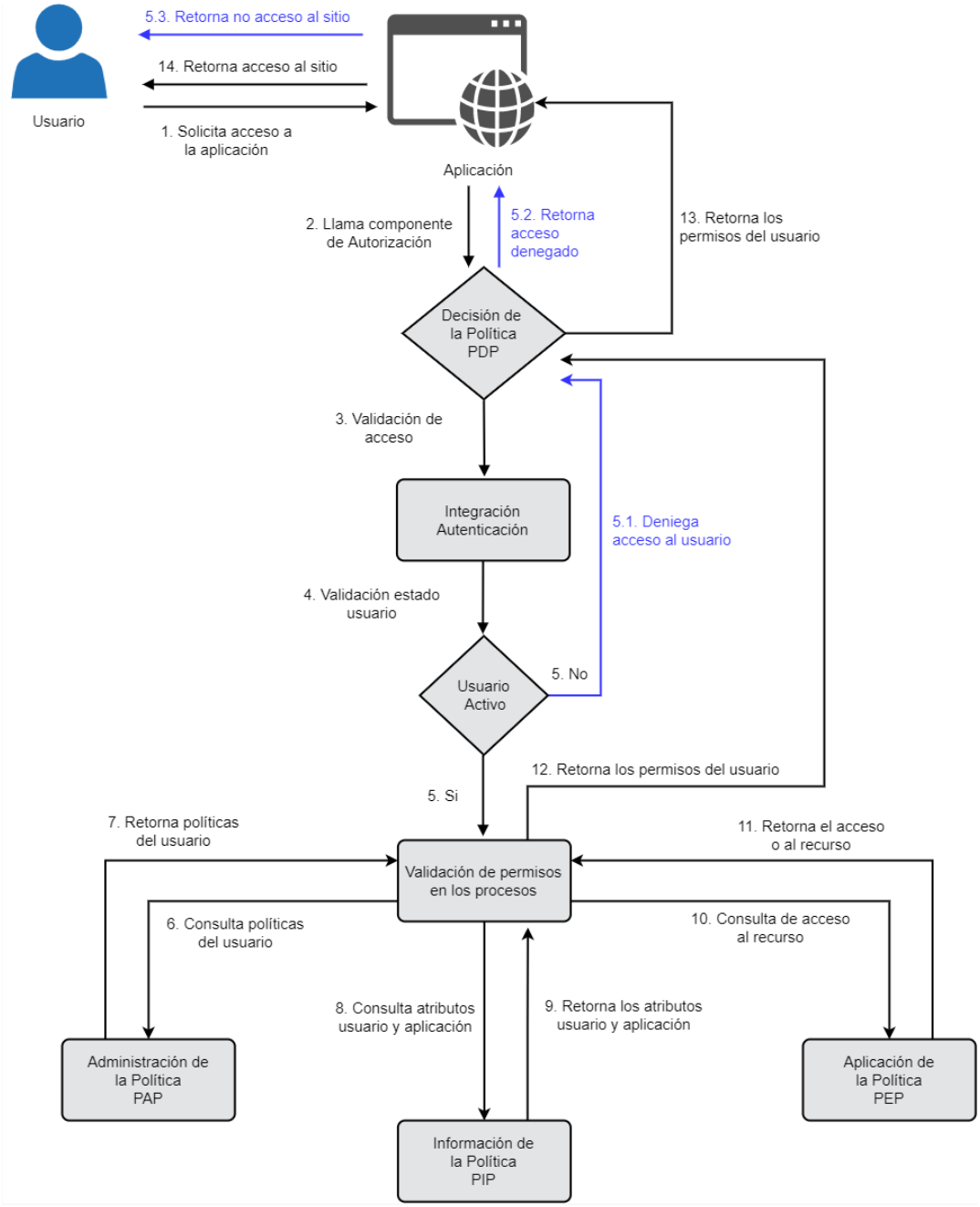
Fuente: propia

En la Figura 3-34 se describió el flujo de información de solicitud de acceso de un usuario a una aplicación y su interacción con los procesos:

1. El usuario envía una solicitud de acceso a la aplicación.
2. La aplicación llama al componente de autorización, inicia el proceso PDP.
3. El PDP llama al componente de Autenticación.
4. El componente de autenticación valida si el usuario está activo o no en la organización.
5. Si no está activo retorna un mensaje de acceso denegado al PDP y este retorna a la aplicación el resultado. De lo contrario el flujo sigue y llama al proceso PAP.
6. Se consulta las políticas del usuario registrado en el PAP.
7. El PAP retorna las políticas configuradas del usuario en la aplicación de negocio como son los roles, niveles de permiso, segregación de funciones.
8. Se consulta al PIP para obtener los atributos del usuario y las configuraciones asignadas a la aplicación de negocio.

9. El PIP envía la información de contexto dinámico de la aplicación al PDP como tipo de dispositivo autorizado, jornada de uso de la aplicación, ubicación geográfica.
10. Se valida al PEP si el usuario puede o no tener acceso al recurso.
11. El PEP retorna la respuesta de acceso al proceso intermedio validación de permisos.
12. Validación de permisos retorna al PDP los resultados encontrados en los procesos PAP, PIP y PEP.
13. El PDP retorna la respuesta de acceso o no a la aplicación.
14. La aplicación le retorna al usuario la respuesta entregada por el PDP.

Figura 3-34. Flujo de información en el modelo centralizado de autorización.



Fuente: Propia.

3.3.3. Componentes del modelo de autorización centralizada

En la Figura 3-35 se indicaron los componentes del modelo centralizado de autorización para que las aplicaciones desarrolladas puedan utilizarlo. Se definieron dos usuarios importantes en este proceso que son el "Administrador de la aplicación de control de acceso centralizado", quien es el encargado de administrar la aplicación de control de acceso, sus parametrizaciones generales y matricular el administrador de la aplicación de negocio, y "Administrador de la aplicación de negocio", quien es el responsable de gestionar el acceso a la aplicación de negocio, definir los roles y niveles de permiso, los atributos de la aplicación.

Los componentes definidos para este modelo son:

- Gestionar Aplicaciones: tiene como propósito listar las aplicaciones que tiene el usuario a su cargo.

- Gestionar Usuarios: Este tiene asignado dos funciones: 1) Asociar los usuarios obtenidos del sistema de autenticación que utiliza la empresa con la aplicación e indicar si está activo o inactivo, y 2) Listar los usuarios asignados a la aplicación.

- Gestionar Niveles de permisos: Tiene asignado dos funciones: 1) Crear los roles que están definidos para la aplicación y si está activo o inactivo, y 2) Listar los roles creados para la aplicación.

- Gestionar Roles: Tiene asignado dos funciones: 1) Crear los roles que están definidos para la aplicación, el nivel de permiso a asignar, si está activo o inactivo., y 2) Listar los roles creados para la aplicación.

- Gestionar Atributos: permite configurar los atributos para la aplicación, es decir, se puede indicar si la aplicación de negocio funciona para ciertas áreas organizacionales, cargos, funciones o roles definidos, así como por ubicación geográfica o por un período de tiempo.

- Gestionar autorización: Tiene asignado dos funciones: 1) Asociar los roles a los usuarios en la aplicación, y 2) Listar los usuarios con sus roles para la aplicación.

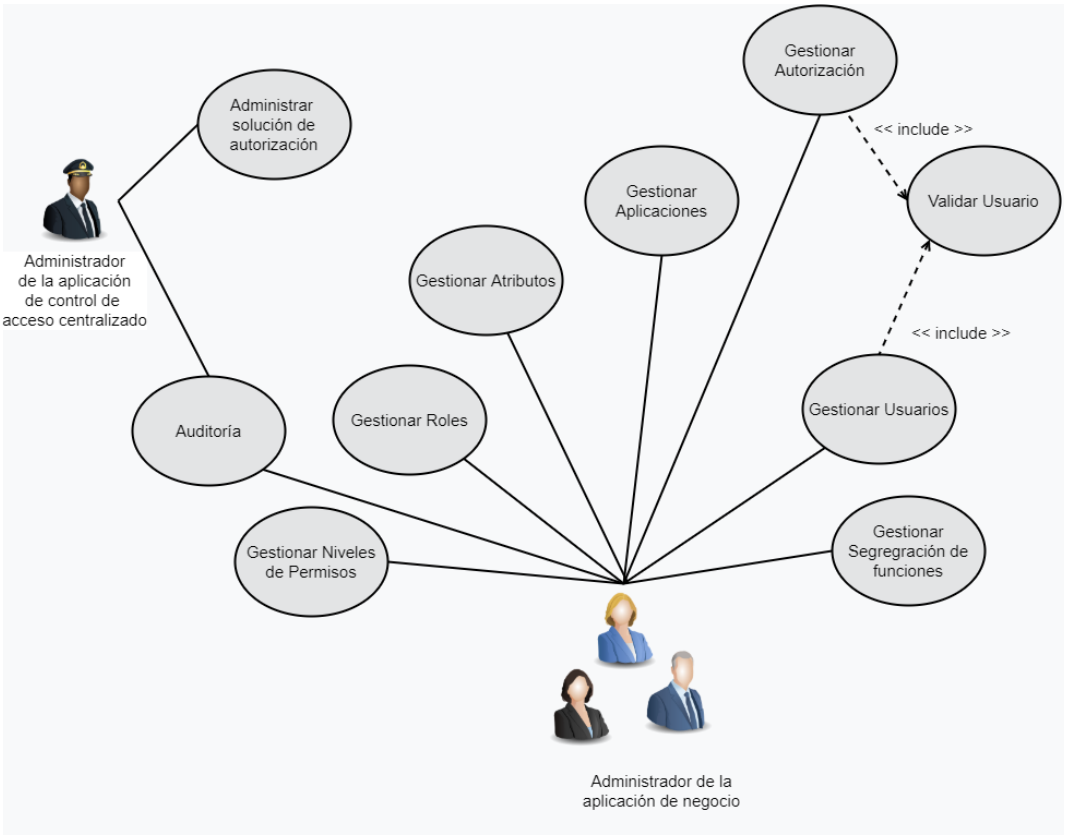
- Gestionar Segregación de funciones: Tiene asignado dos funciones: 1) Agregar los roles que no son compatibles para evitar que sean asignados a los usuarios y evitar abuso de privilegios y 2) Listar las incompatibilidades de permisos definidas en la aplicación.

- Auditoría: Tiene como función poder generar reportes de las configuraciones realizadas a las aplicaciones indicando su fecha, usuario, aplicación, descripción y la acción realizada. También permite verificar el seguimiento de las acciones realizadas por los diferentes administradores tanto de la aplicación de autorización como del negocio, por ejemplo cuándo ingreso al sistema, si cambio el permiso asignado, o cuándo fue retirado. Así como consultar un usuario en qué aplicaciones está asignado y cuáles son sus roles definidos; dependiendo del rol que tenga el usuario logueado en el sistema, la consulta le traerá más o menos información, es decir, si la consulta la realiza el administrador de la aplicación de control de acceso centralizado, éste obtendrá los resultados de todas las aplicaciones donde está el usuario que se desea consultar, pero si la consulta la ejecuta el administrador de la aplicación de negocio, sólo le arrojará la información de las aplicaciones sobre las que él tiene a cargo. Este módulo también permite identificar los permisos y usuarios asignados a una aplicación. Todos estos reportes pueden ser exportados a Excel o pdf.

- Validar Usuario: Es un proceso automático de la solución de autorización para verificar si el usuario al que se le van a asignar los permisos en la aplicación está activo en la organización y se puede continuar con el proceso de asignación, o si se está devolviendo los roles y permisos configurados para ese usuario a la aplicación, antes de enviar la respuesta de validar que si este activo el usuario; en ambos casos, si el usuario esta inactivo devolverá un mensaje indicando que no se pudo realizar la acción solicitada. Así mismo, tiene la tarea de estar verificando si un usuario cambio de rol o está retirado para inactivar los roles asignados.

- Administrar solución de autorización: Es el administrador responsable de la solución centralizada de control de acceso en la organización. Tiene la función de registrar los administradores de las aplicaciones para que ellos puedan configurar su control de acceso y parametrizar las variables que son comunes para todas las aplicaciones como son: nombre de empresa, listado de cargos y ubicación geográfica.

Figura 3-35. Componentes del modelo de autorización centralizada según por usuario.



Fuente: Propia.

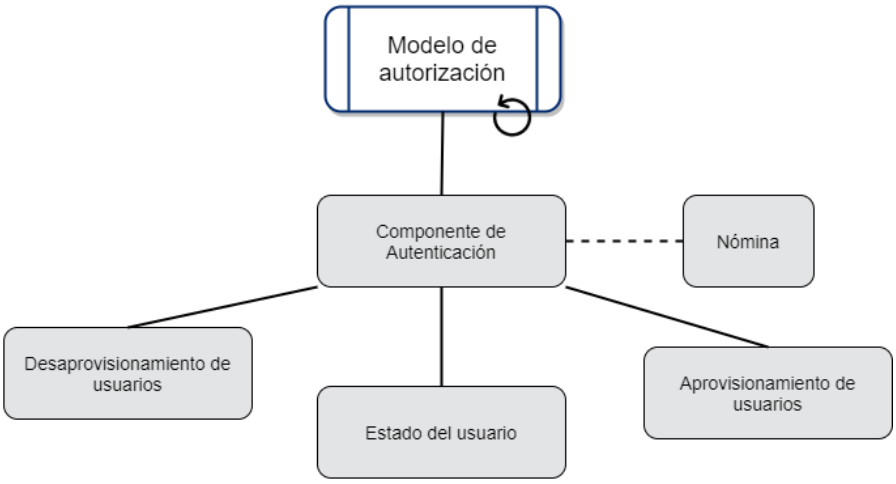
3.3.4. Integraciones del modelo de autorización centralizada

Uno de los aspectos importantes en este modelo es la sincronización automática con el componente de Autenticación, es importante que al momento de retirar un usuario de la organización o cambiar el rol de este, se actualice inmediatamente esta información en el componente de autorización centralizada para evitar fuga de información o uso inapropiado de la aplicación, para ello el componente de autorización debe disponer de un servicio web con los

métodos necesarios para que el componente de Autenticación pueda llamarlos al momento de existir una novedad con un usuario y poder realizarse los ajustes que sean necesarios. Este modelo debe estar continuamente validando con el aprovisionamiento y desaprovisionamiento de usuarios, además, cada vez que vaya a responder sobre el acceso de un usuario hacia la aplicación, debe validar si el éste se encuentra activo o no para poder tomar decisiones sobre si puede o no realizar las funciones respectivas. En caso de encontrar un usuario retirado debe retirar los permisos en las aplicaciones en las que estaba asignado. En la Figura 3-36 se puede apreciar la relación entre el modelo de control de acceso con los demás componentes.

Cuando la organización decide terminar el contrato a un usuario, es importante que la aplicación de nómina envíe de manera oportuna dicho retiro para que el componente de Autenticación pueda notificarle al de Autorización dicho cambio y se retiren definitivamente los permisos al usuario.

Figura 3-36. Integración del modelo de autorización centralizada con otros procesos.



Fuente: Propia

3.3.5. Soluciones de mercado

Se buscaron soluciones en el mercado que fueran líderes en el gobierno y administración de identidades, que en inglés corresponde a *Identity and Access Management* – IGA. Según Gartner, empresa consultora de tecnologías de información que genera un informe cada año sobre las mejores herramientas de mercado, para el año 2019 las empresas líderes en esta área fueron: SailPoint, Saviynt, IBM, Omada y One identity (Figura 3-37). El propósito de estas herramientas es

administrar la identidad digital y los derechos de acceso a las aplicaciones y componentes de TI. Las empresas mencionadas anteriormente tienen tanto ventajas como desventajas, algunas tienen un costo más alto, otras tienen más características funcionales y, sin embargo, estas soluciones son adquiridas por empresas grandes [64].

Figura 3-37. Cuadrante mágico IGA 2019.

Fuente: [64]

Las características comunes de un IGA en las 5 empresas son [64]:

- ✓ Gestión del ciclo de vida de la identidad: corresponde al manejo de la identidad del usuario desde que se crea hasta que se elimina.
- ✓ Gestión de derechos: manejar los derechos de acceso de los usuarios.

- ✓ Solicitudes de acceso: mecanismo de los usuarios para solicitar acceso a los recursos de la organización.
- ✓ Flujos de trabajo: crear flujos para la aprobación de solicitudes como la creación de usuarios, aprobaciones de acceso.
- ✓ Gestión de políticas y roles: se definen y administran las políticas de autorización generadas por las solicitudes de acceso.
- ✓ Auditoría: evaluar las solicitudes y control de acceso generado a los usuarios y generar reportes.

Estas soluciones de mercado no solo tienen el componente de autorización sino un conjunto de características adicionales que conforman un IGA, que no pueden estar independientes, y que las empresas tendrían que adquirir totalmente para usar solo una pequeña parte, gastando recursos económicos innecesarios. Adicional, los requisitos definidos en el modelo de control de acceso centralizado no se logran cumplir a totalidad, debido a que estás soluciones no permiten personalizarse, sino que deben adoptarse como vienen de caja. Por esta razón el modelo aquí propuesto debe ser desarrollado a la medida para que se pueda cumplir con lo especificado y así mismo, permita ser escalable y ajustable en la medida que la organización evoluciona.

3.4. Validación el modelo de seguridad a través de una prueba de concepto, que permita establecer el nivel de cumplimiento en el control de acceso.

Para realizar la validación del modelo propuesto se desarrolló una aplicación donde se incluyeron las historias de usuario que corresponden al producto mínimo viable con el fin de dar cumplimiento al objetivo 4, éstas se detallan en la Tabla 3-31.

Tabla 3-31. Historias de usuario definidas para la validación del modelo.

ID	Título	Funcionalidades	Prioridad
HU-01	Administrar usuarios a la aplicación de negocio	Autorización	Alta
HU-02	Administración de niveles de permisos a la aplicación de negocio	Autorización	Alta
HU-06	Administración de roles a la aplicación de negocio	Autorización	Alta
HU-07	Administración de atributos a la aplicación de negocio	Autorización	Alta
HU-13	Registrar aplicación de negocio para control de acceso centralizado	Administración	Alta
HU-26	Aplicaciones asignadas al responsable de la aplicación de negocio	Autorización	Alta
HU-09	Segregación de funciones	Autorización	Media
HU-17	Cierre de sesión a la aplicación de negocio	Administración	Media
HU-10	Sistema de control de acceso multifilial	Autorización	Baja

3.4.1. Construcción de la aplicación

Para hacer el desarrollo se utilizaron las siguientes herramientas:

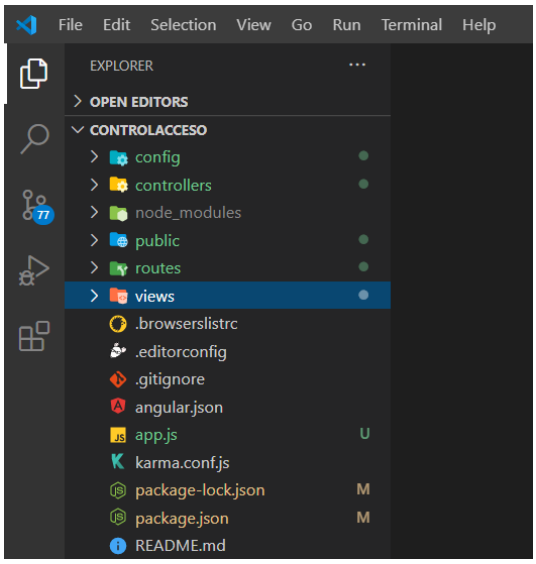
- ✓ Visual Studio Code - VSC: Editor de código fuente con versión 1.51.1.
- ✓ NodeJS: Corresponde a la capa del servidor basado en el lenguaje de programación JavaScript, la versión instalada fue 12.18.4.
- ✓ MySQL: Motor de base de datos y su versión fue 8.0.21.
- ✓ Se instalaron los siguientes módulos en NodeJS como apoyo para el desarrollo: express, express-handlebars, express-mysql-session, mysql, dotenv, hbs, connect-flash, bcryptjs, jsonwebtoken y cookie-parser.

Al ser un prototipo para la validación del modelo, este desarrollo no se integró a un directorio activo sino que se creó su propio componente de autenticación con el propósito de poder llevar a cabo la prueba de concepto, es de aclarar, que el modelo al implementarse en una empresa si debe estar integrado para obtener mejores beneficios.

En la Figura 3-38 se puede observar cómo se estructuró el proyecto en Visual Studio Code, se identificaron los siguientes componentes más importantes:

- ✓ config: donde se encuentran las variables de configuración de conexión hacia la base de datos y de más parámetros que son transversales a la aplicación.
- ✓ controllers: la lógica de programación según la ruta indicada por el usuario.
- ✓ public: los archivos que son públicos como las imágenes, javascripts y las hojas de estilo para que el sitio sea agradable en su presentación.
- ✓ routes: se definen las rutas de navegación (URI - Uniform Resource Identifier) y que acción debe ejecutar en el controllers según la elección del usuario.
- ✓ views: corresponden a las vistas o páginas web de la aplicación.
- ✓ app.js: Es el archivo de inicio de la aplicación donde se indican las configuraciones transversales como la conexión a la base de datos, el manejo de la sesión, variables globales, los archivos de routes que se van a utilizar y el puerto por el que va a escuchar la aplicación.

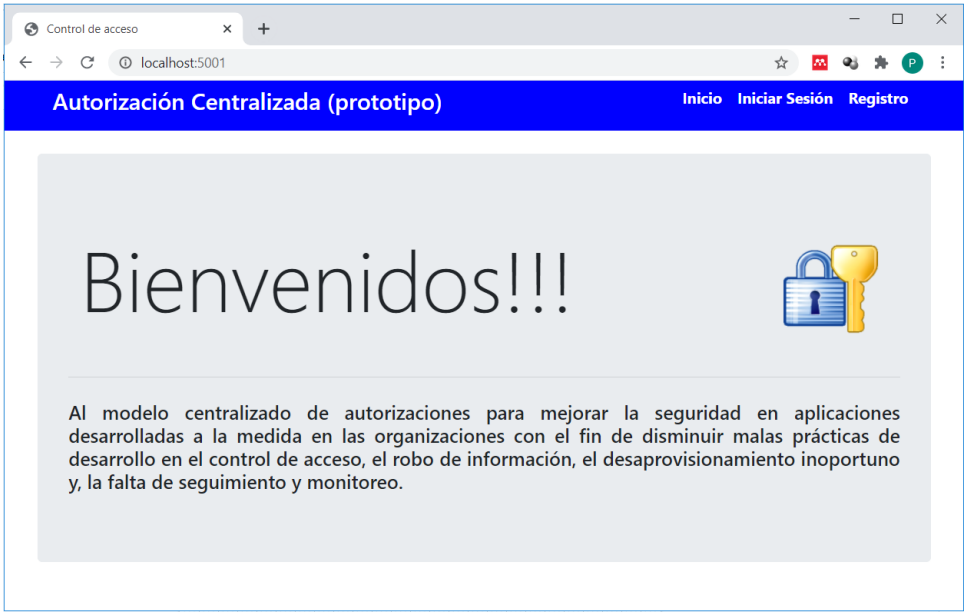
Figura 3-38. Estructura del proyecto en VSC



Fuente: Propia.

La página principal de la aplicación se encuentra en la Figura 3-39, allí el usuario tiene las opciones de Iniciar Sesión o Registrarse y se le indica una breve descripción del propósito de ésta.

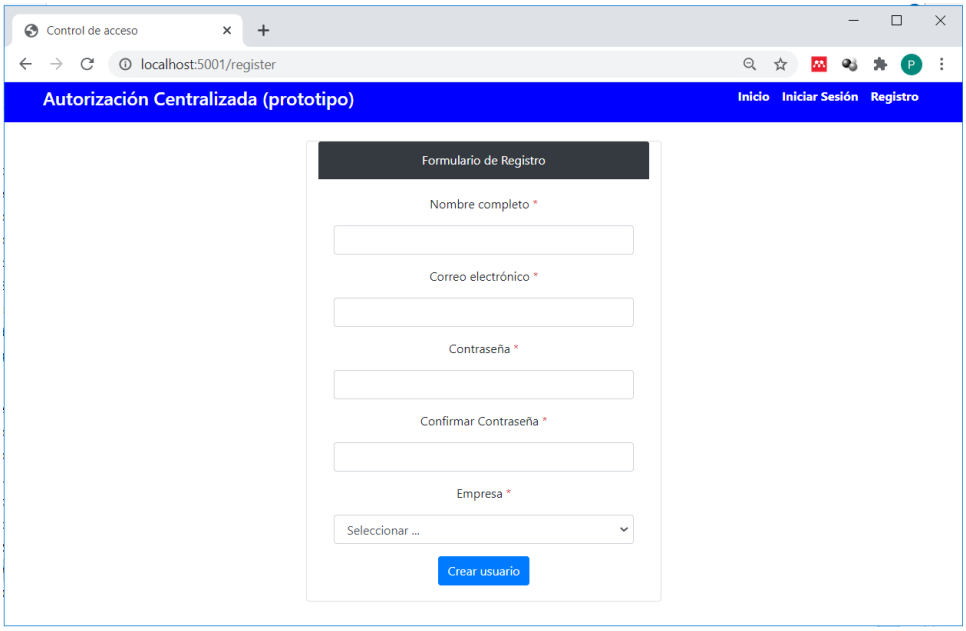
Figura 3-39. Página de inicio de la aplicación.



Fuente: Propia.

Para registrar o crear un usuario se debe diligenciar el nombre, correo electrónico, contraseña y la empresa a la cual pertenece, en la Figura 3-40 se encuentra el diseño utilizado para la construcción de este formulario.

Figura 3-40. Formulario de registro en la aplicación.



Fuente: Propia

Para el formulario de inicio de sesión, se solicita un correo electrónico y una contraseña con el fin de hacer la validación de si existe o no, durante esta parte puede ocurrir cuatro situaciones:

- ✓ Correo no registrado: el cual muestra el mensaje: "El usuario no está registrado" al usuario indicando que no existe en la aplicación.
- ✓ Correo o contraseña incorrectas: el cual muestra el mensaje: "Correo o contraseña incorrecta" al usuario indicando que revise sus datos ingresados para poder iniciar sesión.
- ✓ Usuario inactivo: el cual muestra el mensaje: "Usuario inactivo" al usuario indicando que el usuario no puede acceder a la aplicación.
- ✓ Correo y contraseña válidas: la aplicación de autorización muestra los menús disponibles de acuerdo con el rol asignado en la aplicación.

3.4.2. Módulos del modelo de autorización centralizada

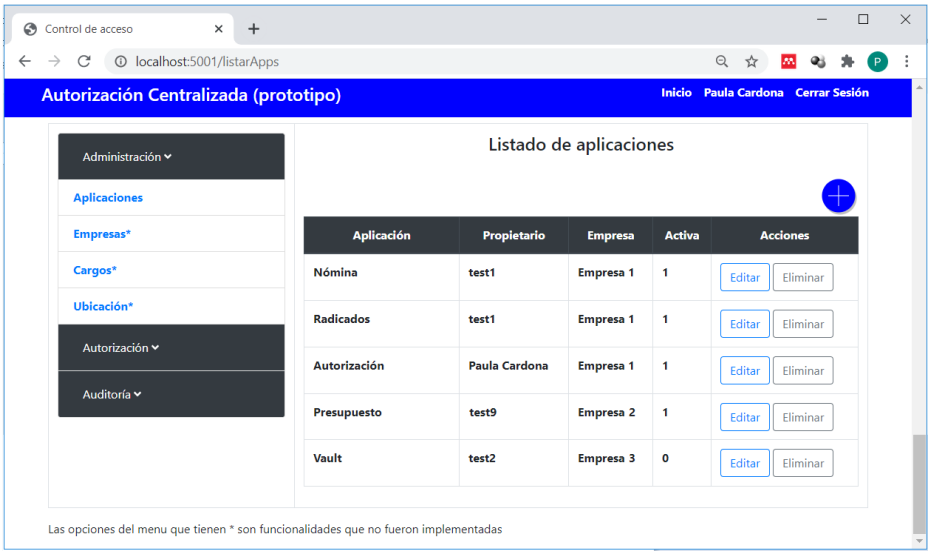
De acuerdo con lo definido en el modelo centralizado de autorización, se acordaron 3 funcionalidades:

- **Módulo de Administración**

El cual es gestionado por el administrador del componente de autorización y puede llevar a cabo las siguientes tareas que están disponibles en el menú lateral izquierdo como se ilustra en la Figura 3-41:

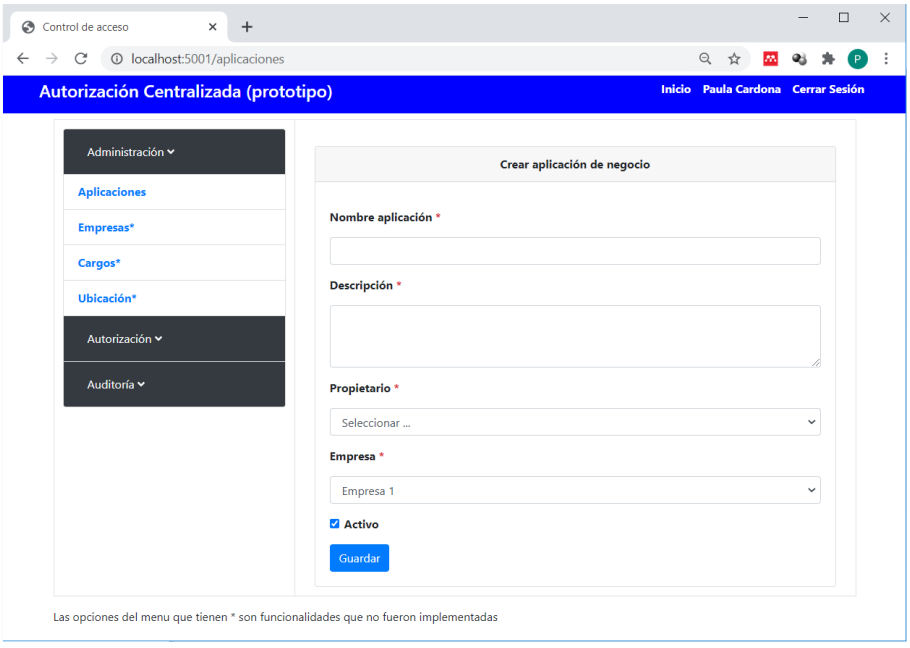
- ✓ Aplicaciones: donde se puede registrar la aplicación de negocio para una empresa en particular y asignar los responsables de éste. Un ejemplo del formulario para crear la aplicación de negocio se puede ver en la Figura 3-42, allí se guarda el nombre, la descripción, el responsable y la empresa a la que pertenece. Por ser piloto se acordó que solo un administrador por aplicación, pero la propuesta indica que debe ser existir mínimo dos responsables.
- ✓ Empresas: Donde se administran las empresas a las cuales la aplicación de autorización prestará su servicio. No fue implementado en el piloto, solo se crearon las tablas en la base de datos para lograr el funcionamiento del prototipo.
- ✓ Cargos: Se registra la estructura organización de cada empresa que utiliza esta aplicación. No fue implementado en el piloto, solo se crearon las tablas en la base de datos.
- ✓ Ubicación: Se ingresan las diferentes ubicaciones tales como país, departamento, ciudad, municipio o sector. No fue implementado en el piloto, solo se crearon las tablas en la base de datos.

Figura 3-41. Menú del administrador y formulario de listar aplicaciones.



Fuente: Propia

Figura 3-42. Formulario para crear aplicación de negocio.



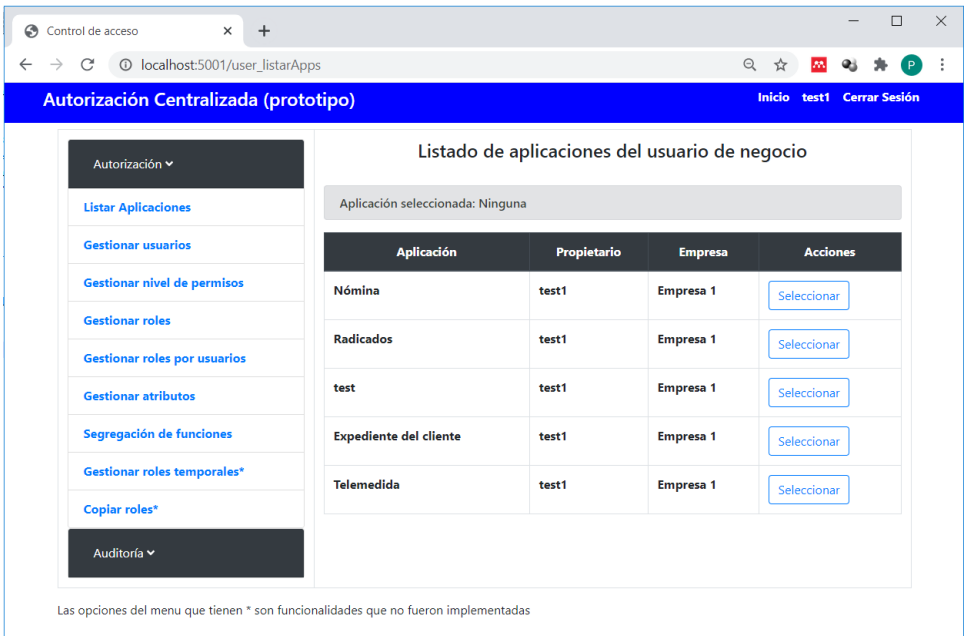
Fuente: Propia.

- **Módulo Autorización**

Esta funcionalidad de la aplicación de autorización centralizada permite al administrador de la aplicación de negocio realizar las siguientes tareas:

✓ Listar Aplicaciones: En esta sección el usuario podrá visualizar las aplicaciones que tiene a cargo y seleccionar alguna para administrar su control de acceso. La Figura 3-43 es un ejemplo de cómo se visualizan las aplicaciones, se muestra el nombre de la aplicación, el propietario, la empresa a la que pertenece y la acción de seleccionar.

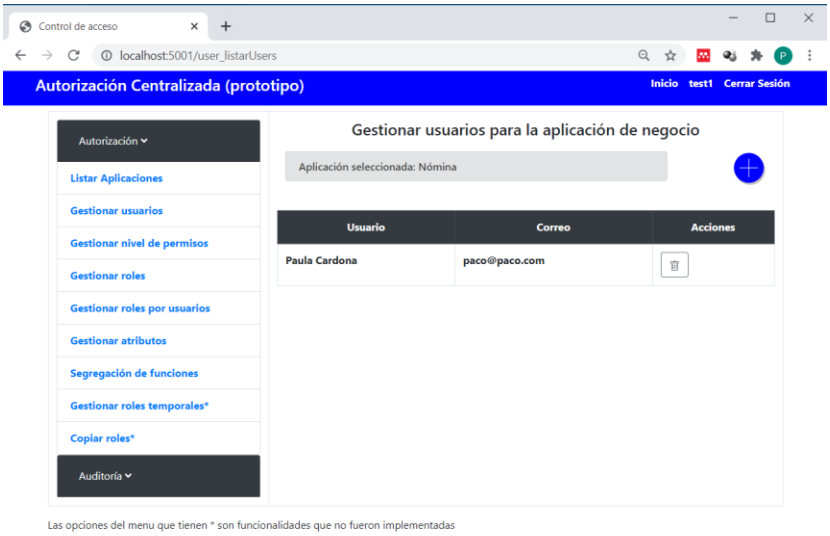
Figura 3-43. Listado de aplicaciones del administrador de la aplicación de negocio.



Fuente: Propia.

✓ Gestionar usuarios: el administrador de la aplicación de negocio podrá registrar o retirar los usuarios que van a hacer uso de la aplicación. Un ejemplo de esto se puede observar en la Figura 3-44, donde se definió una tabla con las columnas: usuario, correo, la acción de eliminar, y el botón para agregar a alguien más.

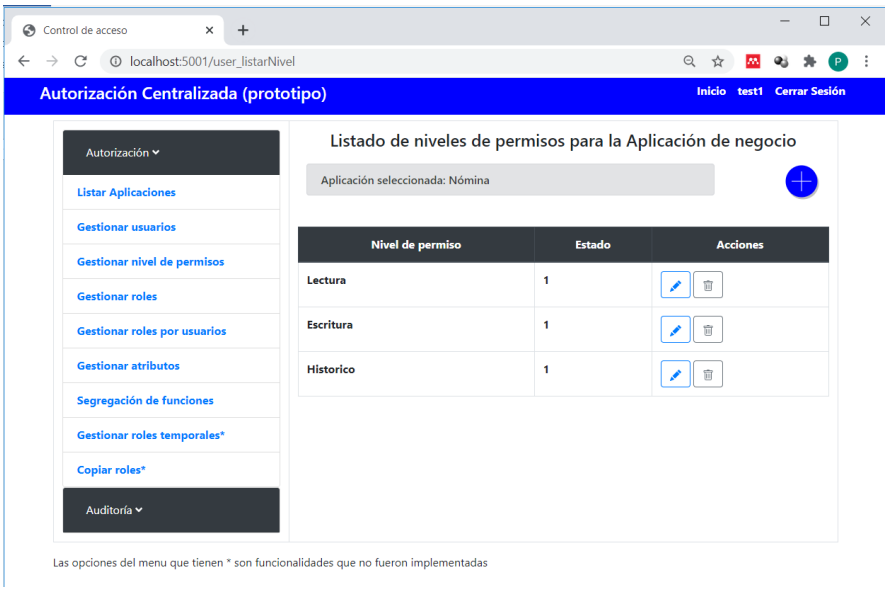
Figura 3-44. Gestionar usuarios en la aplicación de negocio.



Fuente: Propia

✓ Gestionar nivel de permisos: el usuario con los permisos respectivos podrá registrar o eliminar las acciones u operaciones que se puede realizar en la aplicación de negocio. Por ejemplo: crear, eliminar, leer o modificar. En la Figura 3-45 se encuentra el diseño propuesto para esta tarea, donde se listan los niveles de permiso, su estado (activo o inactivo), las acciones de editar o eliminar y el botón para crear un nuevo registro.

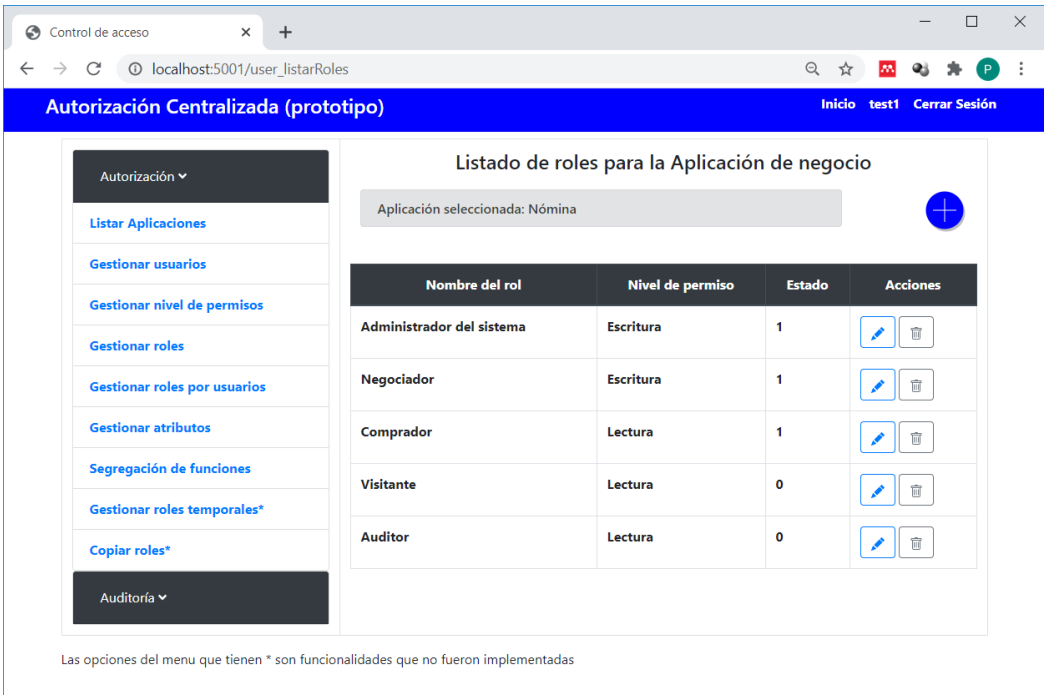
Figura 3-45. Listar los niveles de permiso para la aplicación de negocio.



Fuente: Propia.

- ✓ Gestionar roles: sección donde se muestran o se crean los roles o funciones definidas para una aplicación de negocio y se asocia al nivel de permiso definido previamente. Para la implementación en el piloto, se definió un rol por un nivel de permiso, al momento de usarse en una organización, se puede desarrollar para que un rol pueda tener más de un nivel de permiso. En la Figura 3-46 se listan los roles ejemplo para la aplicación Nómina, el nivel de permiso asignado a cada rol, su estado (activo o inactivo), las acciones para editar o eliminar y el botón para agregar uno nuevo.

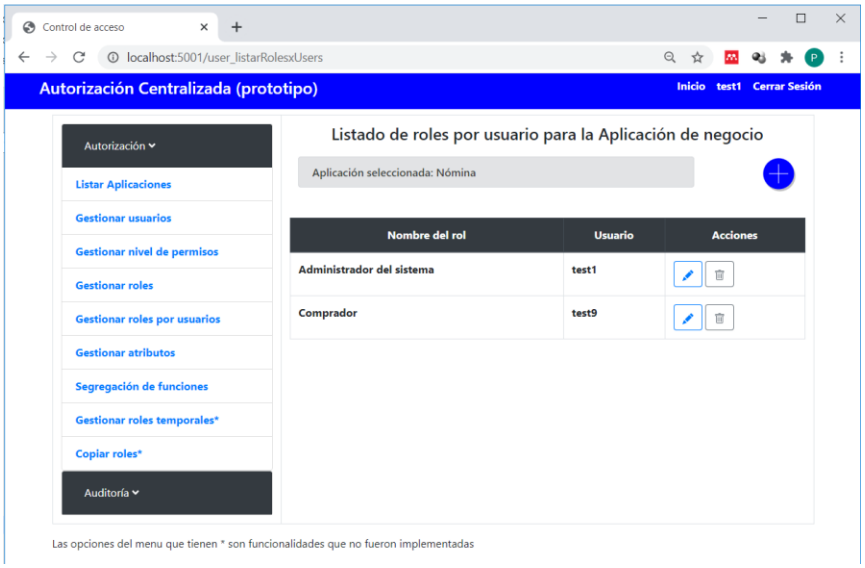
Figura 3-46. Gestionar los roles para la aplicación de negocio.



Fuente: Propia.

- ✓ Gestionar roles por usuarios: después de tener configurado los roles, el paso siguiente es asignarles a los usuarios registrados para la aplicación los roles o funciones que van a llevar a cabo en la aplicación de negocio. El formulario del prototipo desarrollado muestra el rol, el usuario asignado a éste, las acciones de editar o eliminar y el botón para agregar más registros, esto se puede observar en la Figura 3-47.

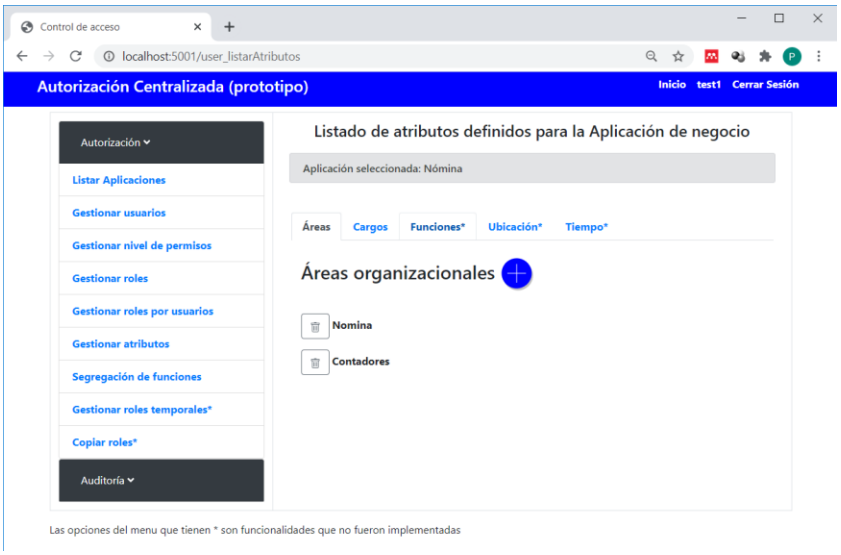
Figura 3-47. Gestionar roles por usuarios para la aplicación de negocio.



Fuente: Propia.

- ✓ Gestionar atributos: En esta sección, se configura las restricciones de uso de la aplicación según el entorno del usuario, en la Figura 3-48 se observan los atributos de Área organizacional, cargos, funciones, ubicación y tiempo. Para el ejemplo de la aplicación Nómina, se configuró que solo las áreas de Nómina y Contadores pueden acceder a ésta.

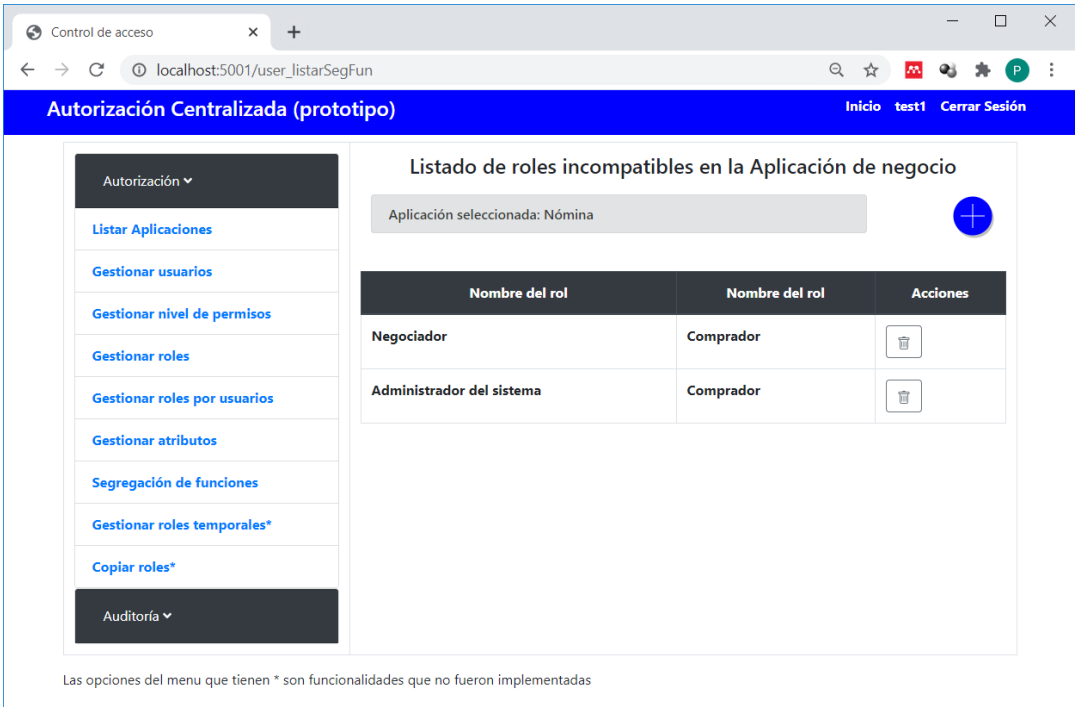
Figura 3-48. Atributos de la aplicación de negocio.



Fuente: Propia.

✓ Segregación de funciones: el administrador de la aplicación de negocio puede en esta sección indicar cuáles roles no son compatibles para que sean asignados a un usuario. Al momento de otorgarle el permiso a este, se realiza la validación y si se encuentra registrada alguna inconformidad en la segregación de funciones, esta no es concedida. En la Figura 3-49 se muestra un ejemplo de incompatibilidad de roles para la aplicación Nómina, un usuario no puede tener al mismo tiempo los roles de Negociador y Comprador. Es importante aclarar que la incompatibilidad de roles debe ser suministrada por el dueño de la aplicación de negocio, no es responsabilidad de este componente hacerlo, éste solo provee la herramienta para hacer cumplir la segregación de funciones.

Figura 3-49. Segregación de funciones para la aplicación de negocio.



Fuente: Propia.

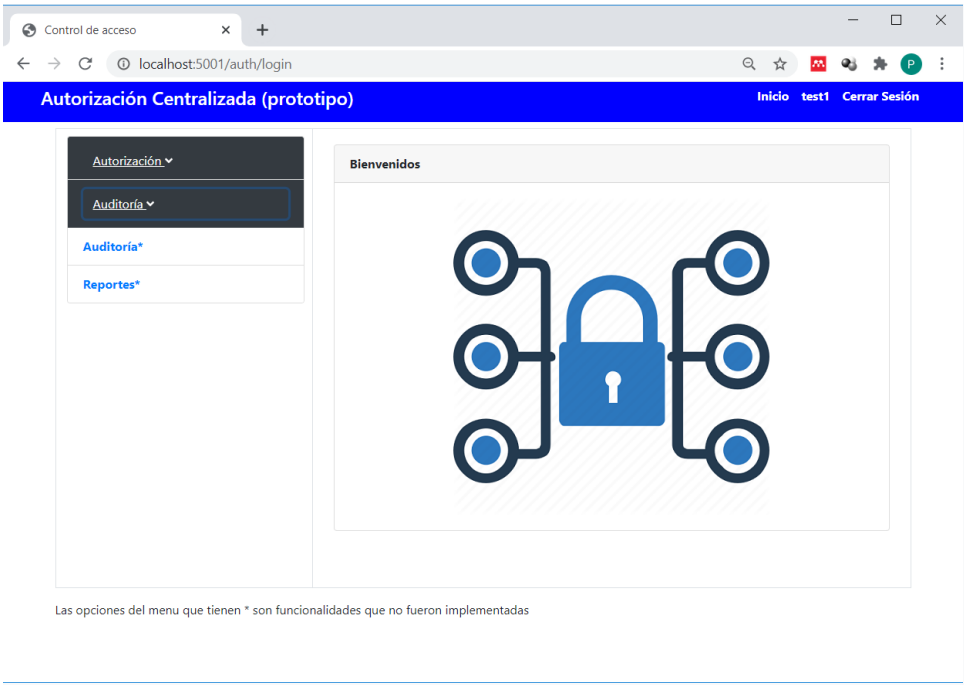
✓ Gestionar roles temporales: Donde se asignan por un período de tiempo un permiso a un usuario en la aplicación de Negocio. Esto puede ser utilizado cuando existe una incapacidad, vacaciones, licencias entre otros. Para el piloto, esta sección no fue implementada.

- ✓ Copiar roles: Le permite al administrador de la aplicación de negocio copiarse la estructura de permisos de una aplicación para ser replicada en otra, esto facilita su gestión en caso por ejemplo que la aplicación deba ser clonada porque será utilizada en una filia. Para el piloto, esta sección no fue implementada.

- **Módulo de auditoría**

Esta funcionalidad de la aplicación de autorización centralizada permite al administrador de la aplicación de negocio o del componente de autorización realizar seguimiento a los permisos otorgados, conocer la configuración de control de acceso de una aplicación o incluso, identificar un usuario a que aplicaciones pertenece y cuáles permisos le fueron otorgados. Para el piloto, esta sección no fue implementada. En la Figura 3-50 se muestra un ejemplo de cómo sería el menú para acceder a estas tareas definidas a los usuarios, donde se encuentra la parte de Auditoría y Reportes.

Figura 3-50. Auditoría y seguimiento al control de acceso en la aplicación de negocio.



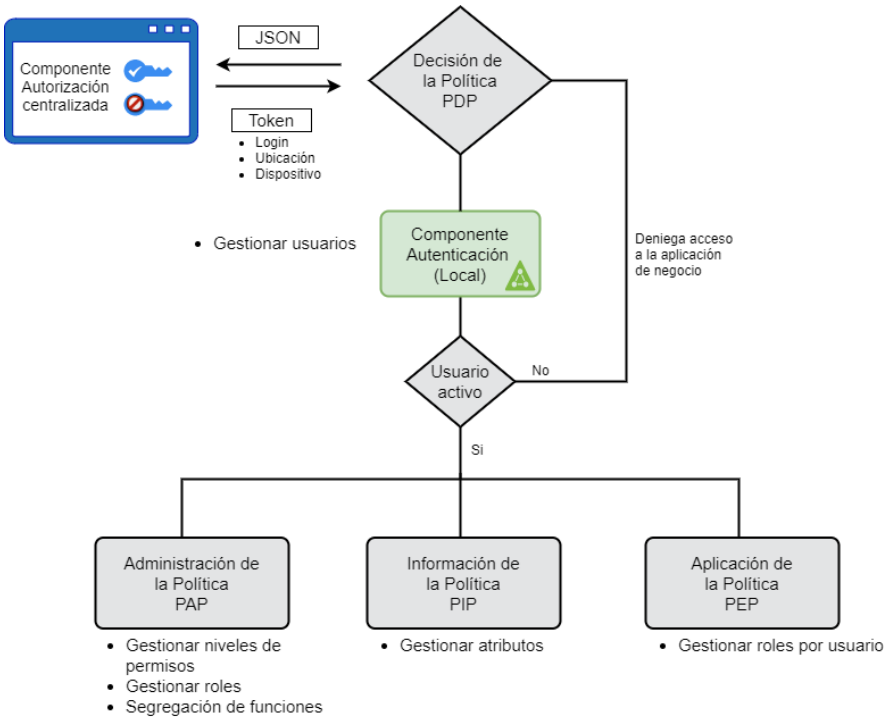
Fuente: Propia.

- **Modelo centralizado versus aplicación desarrollada**

De acuerdo con el desarrollo realizado para la validación del modelo centralizado de autorización, se puede observar en la Figura 3-51 cada una de las secciones del módulo Autorización como se relacionan entre sí, los otros dos módulos no son incluidos porque son soporte para su funcionamiento. A continuación, se describen los procesos:

a. Componente Autenticación: Se utilizó una autenticación local, donde se definió un atributo para indicar si el usuario está activo o no, dependiendo del resultado dejará ingresar o no a la aplicación del componente de autorización, esto con el fin de validar el comportamiento de un usuario activo y uno inactivo y su respuesta.
b. PAP: Allí se configuraron las secciones de "Gestionar niveles de permisos", "Gestionar roles" y "Segregación de funciones".
c. PIP: Se define la sección de "Gestionar atributos".
d. PEP: Se indica si el usuario tiene o no acceso con la sección "Gestionar roles por usuario".

Figura 3-51. Modelo centralizado de autorización versus aplicación desarrollada



Fuente: Propia

En la siguiente Tabla 3-32 se muestra la relación de lo desarrollado en el piloto con cada uno de los modelos de autorización utilizados para la definición del modelo centralizado.

Tabla 3-32. Modelos de autorización versus elementos relacionados.

Modelo de autorización	Elementos relacionados
RBAC	• Nivel de permisos • Roles • Roles por usuario • Segregación de funciones
ABAC	• Áreas • Cargos • Funciones • Ubicación
CBAC	• Tiempo

Se debe aclarar que la aplicación desarrollada no tiene todos los elementos definidos en el modelo. Algunos son:

- ✓ La integración con el componente de autenticación; para este caso se utilizó una autenticación local, donde se creó el campo *Enabled* de tipo booleano para indicar si el usuario está activo o no.
- ✓ Para los atributos de tipo contexto solo se utilizó el tiempo, pero no se agregaron otros como dirección ip, tipo de dispositivo, entre otros.
- ✓ No se desarrolló el mecanismo de respuesta a la aplicación de negocio en formato json, ni los servicios web que se deben exponer para su consumo.

3.4.3. Resultado de las entrevistas acerca de la evaluación del modelo

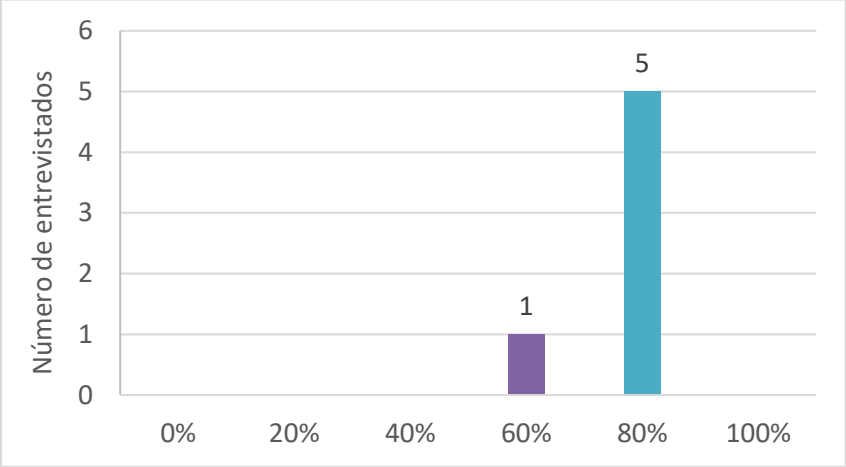
Como se indicó en la metodología, se realizaron seis entrevistas a personas del área de TI, las cuales habían sido previamente entrevistadas en el objetivo 1, con el propósito de mostrarles el modelo diseñado y construido parcialmente para la centralización de autorización en las aplicaciones

desarrolladas a la medida y proceder a evaluarlo e indicar si éste satisface la mayoría de los problemas planteados en este proyecto de grado y su pertinencia en las organizaciones.

A continuación, se exponen los doce resultados obtenidos por cada una de las preguntas evaluadas a los entrevistados sobre el modelo propuesto:

a. Para la pregunta: *Si tuviera el modelo propuesto implementado en la organización con una política de control de acceso a nivel corporativo y, procedimientos claros, definidos, divulgados y revisados periódicamente para las aplicaciones desarrolladas a la medida. ¿En qué porcentaje le ayudaría a disminuir los problemas planteados a su empresa?*, las respuestas de los entrevistados se pueden observar en la Figura 3-52 . Se logró determinar que en un 80% el modelo les ayuda a reducir los problemas frente al control de acceso mitigando el riesgo de eliminación, modificación o fuga de información por personal no autorizado.

Figura 3-52. Satisfacción del modelo propuesto en la organización.

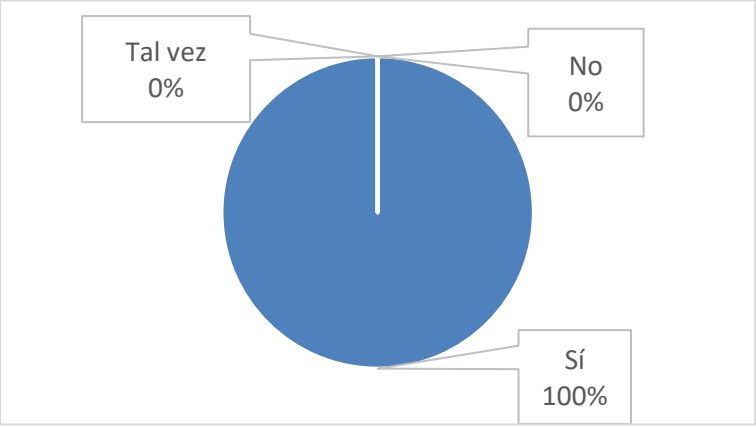


Fuente: Propia.

b. Frente a la inquietud: *Si implementara el modelo centralizado de autorización, ¿consideraría que valdría la pena disponer de un solo mecanismo de control de acceso y no un repositorio mixto donde algunas aplicaciones manejen su propio mecanismo?*, en las respuestas de los entrevistados, se puede observar que el 100% está de acuerdo con utilizar un solo mecanismo de autorización en toda la organización, para más detalle ver la Figura 3-53. En sus comentarios

indicaban que esto facilita la administración y control de los usuarios en las aplicaciones y la integración con los sistemas como directorio activo permite una sincronización automática disminuyendo el soporte por parte de los analistas de seguridad en las aplicaciones.

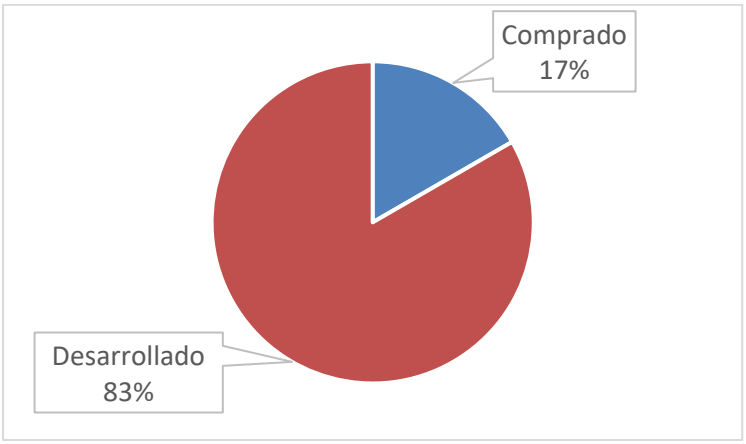
Figura 3-53. Recomendación del mecanismo de autorización centralizado.



Fuente: Propia.

c. En la consulta: *Si usted tuviera la posibilidad de elegir entre buscar en el mercado una herramienta de control de acceso y el desarrollo de este modelo ajustado a las necesidades de su organización, ¿Cuál elegiría?,* las respuestas se pueden observar en la Figura 3-54. El 83% de los entrevistados manifestaron que prefieren desarrollar la aplicación de control de acceso centralizado porque permite personalizar sus necesidades.

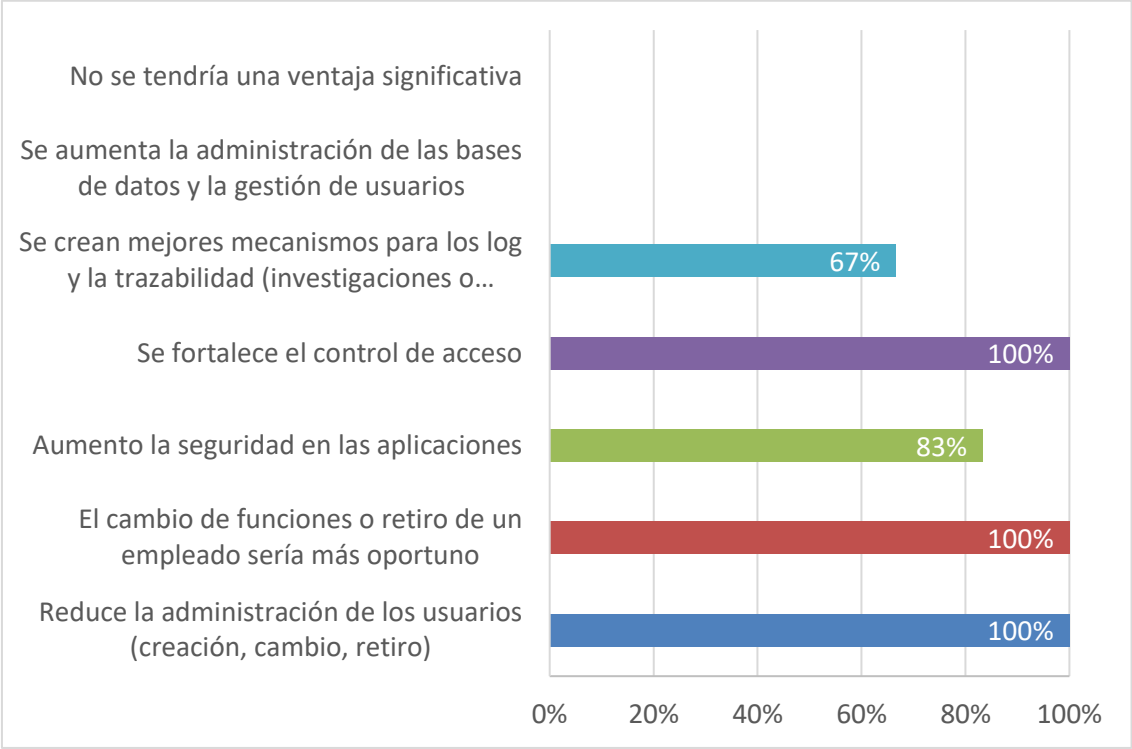
Figura 3-54. Componente de autorización comprado o desarrollado.



Fuente: Propia.

d. Para la siguiente pregunta: *Si tuviera implementado este modelo, ¿Cuáles de los siguientes elementos le generaría beneficios para su organización?*, en la Figura 3-55 se pueden observar los resultados. Se concluyó que todos los entrevistados están de acuerdo con que se fortalece el control de acceso, el cambio de funciones o retiro de un empleado es más oportuno y se reduce la administración de la gestión de los usuarios en la aplicación. El 83% encuentran beneficio que se mejora la seguridad en las aplicaciones. En general, todos consideran que son más los beneficios de disponer de un control de acceso centralizado versus uno descentralizado.

Figura 3-55. Ventajas del modelo propuesto centralizado.

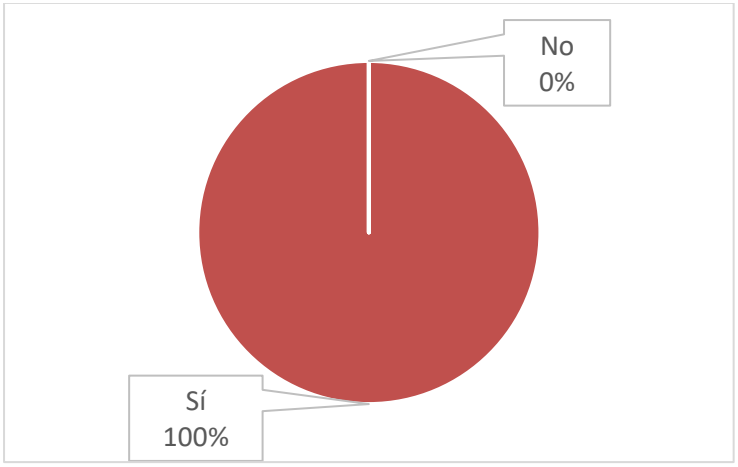


Fuente: Propia.

e. Con respecto a la cuestión: *El modelo propuesto está basado en RBAC (Role Based Access Control), ABAC (Attribute Based Access Control) y CBAC (Context Based Access Control), ¿Considera que es suficiente o podría adicionarse otro mecanismo de control de acceso? ¿Cuál sería?,* según la Figura 3-56 se muestra que todos los entrevistados están de acuerdo o

consideran que los tres modelos de control de acceso para las aplicaciones es suficiente para mejorar la seguridad en las aplicaciones desarrolladas a la medida.

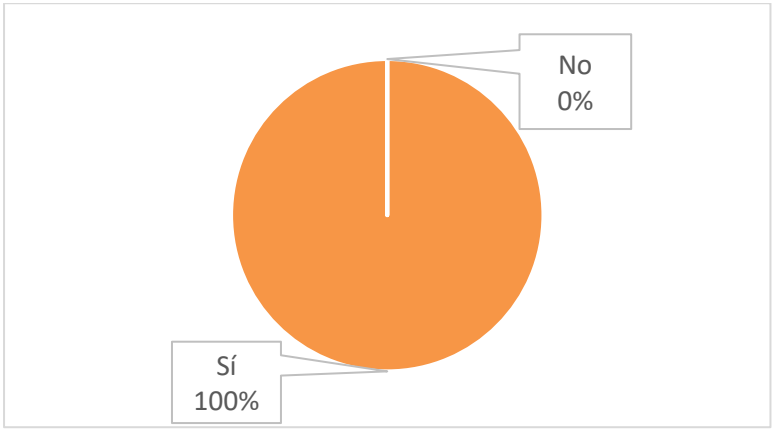
Figura 3-56. Modelo construido basado en RBAC, ABAC y CBAC.



Fuente: Propia.

f. Frente a la pregunta: *El modelo propuesto contempla que su implementación utilice el estándar Oauth debido a que las aplicaciones tienden a estar en la nube. ¿Considera que es suficiente o podría utilizarse otro estándar? ¿Cuál sería?* De acuerdo con la Figura 3-57, los entrevistados coinciden con que el mecanismo o protocolo Oauth es recomendable para esta implementación del modelo centralizado de autorización.

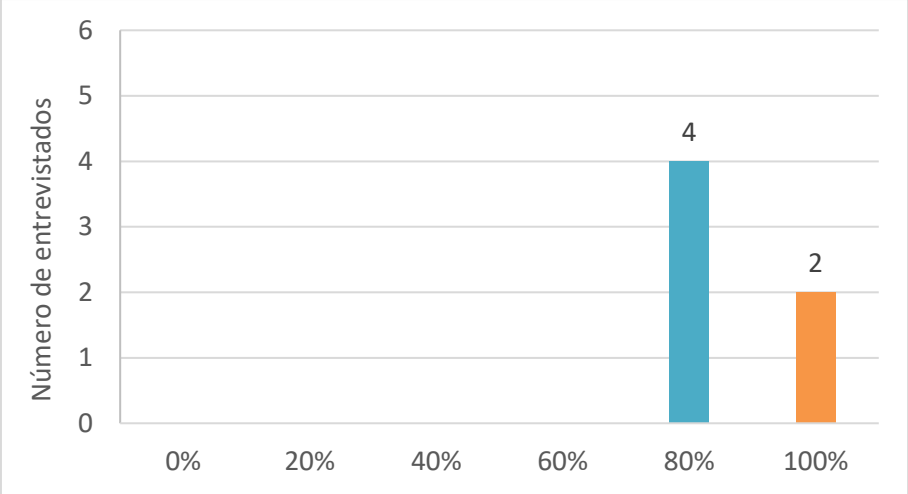
Figura 3-57. Recomendación de utilizar Oauth.



Fuente: Propia.

g. Con respecto a la inquietud: *Con este modelo centralizado de autorización implementado en la organización para las aplicaciones desarrolladas a la medida, con un bueno gobierno y procedimientos claros ¿En qué porcentaje consideraría usted que desde la etapa de levantamiento de requisitos se definiría el control de acceso?* Como aparece en los resultados arrojados en la Figura 3-58, algunos de los entrevistados piensan que en un 80% este modelo apoya en parte a mejorar el control de acceso de las aplicaciones desarrolladas a la medida desde la etapa de requisitos. Y el resto considera que lo hace en un 100% a la organización.

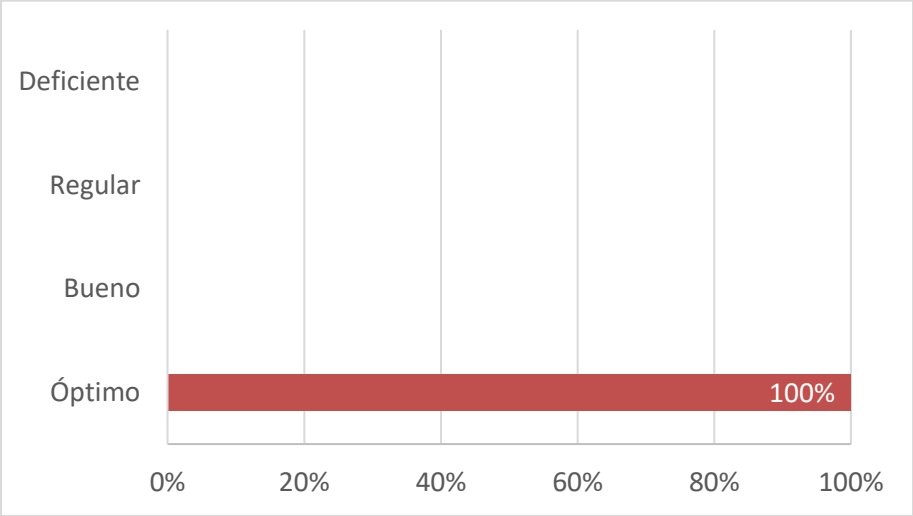
Figura 3-58. Modelo apoya al levantamiento de requisitos.



Fuente: Propia

h. Para la pregunta: Con este modelo centralizado de autorización implementado en la organización ¿qué tan efectivo considera que sería la inactivación o modificación de los perfiles cuando una persona cambia de funciones o se retira de la empresa? Según la Figura 3-59, todos los entrevistados indicaron que este modelo es óptimo para la inactivación o modificación de perfiles por la sincronización con el componente de autenticación y el manejo de atributos en la aplicación de negocio.

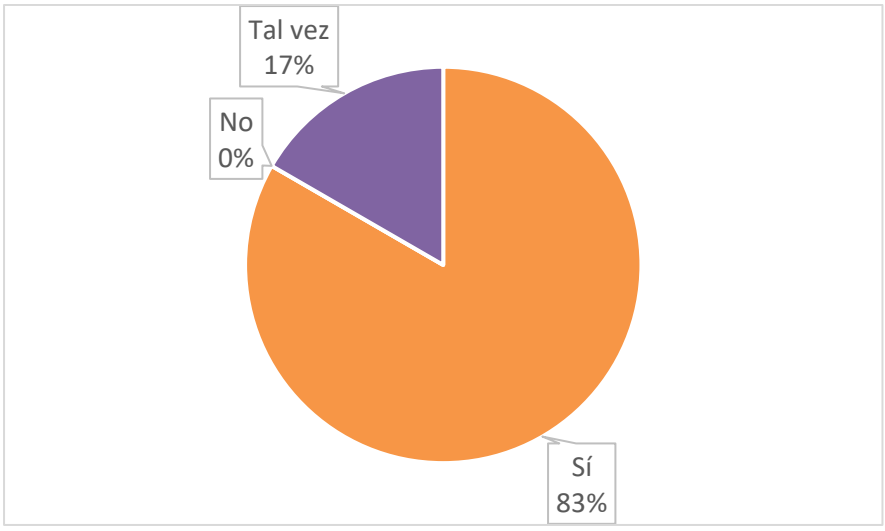
Figura 3-59. Efectividad en la inactivación de los roles en la aplicación de negocio.



Fuente: Propia.

i. En la consulta: *¿Con este modelo consideraría qué se puede ayudar al principio del mínimo privilegio para la gestión de accesos y apoyaría al proceso de segregación de funciones para evitar fraudes entre otros?,* como se muestra en la Figura 3-60, el 83% de los entrevistados respondieron que sí, que este modelo puede ayudar al principio del mínimo privilegio en las aplicaciones.

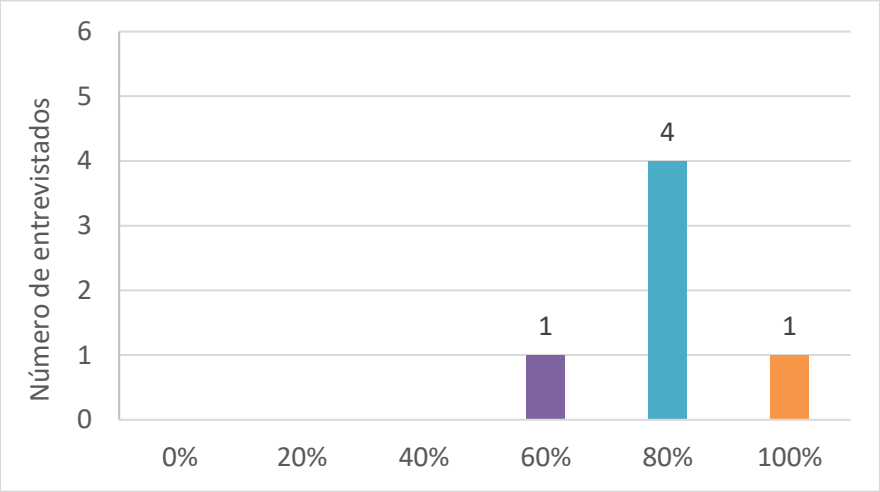
Figura 3-60. El modelo apoya al principio del mínimo privilegio.



Fuente: Propia.

j. Frente a la pregunta: *Con este modelo centralizado de autorización al ser desarrollado, permite que la administración del control de acceso se haga de manera automática y manual. ¿En qué porcentaje consideraría usted que le estaría ayudando a reducir el soporte de la seguridad en su organización y ser más efectivo en su asignación automática?* Se puede observar en la Figura 3-61, que los entrevistados indican que el 80% del modelo les ayuda a disminuir el soporte durante la asignación o retiro de los permisos de un usuario en la aplicación.

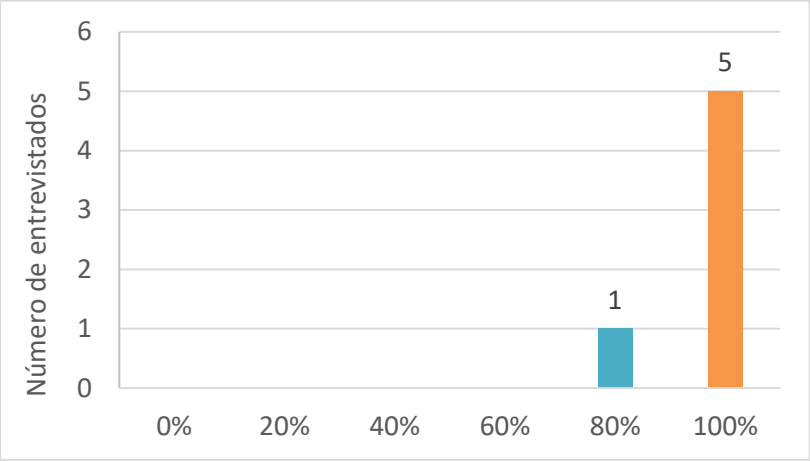
Figura 3-61. Reducción de soporte en la atención del control de acceso.



Fuente: Propia.

k. Para la cuestión: *Este modelo centralizado de autorización permite que al retirarse un empleado de la empresa, los permisos asignados en las aplicaciones desarrolladas a la medida se eliminen de manera automática. ¿En qué porcentaje consideraría usted que esto le aportaría beneficios a su organización?* De acuerdo con la Figura 3-62, la mayoría de los entrevistados coinciden que el modelo les arroja beneficios de un 100% en la gestión del control de acceso en las aplicaciones desarrolladas a la medida.

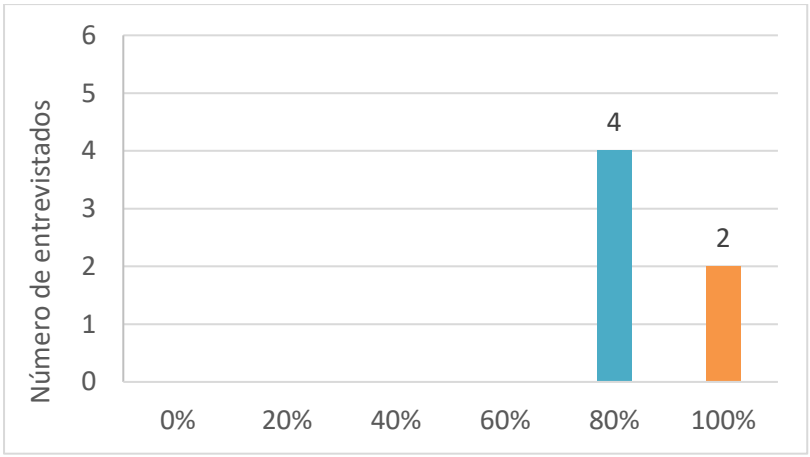
Figura 3-62. Efectividad en el retiro de los permisos de forma automática.



Fuente: Propia

l. Con respecto a la duda: *Con este modelo implementado se puede hacer seguimiento, monitoreo y trazabilidad de los permisos asignados de un usuario en una o varias aplicaciones desarrolladas a la medida, incluso al ser un desarrollo se puede implementar el componente para que envíe periódicamente el reporte de los accesos que tiene parametrizado una aplicación para su revisión ¿En qué porcentaje consideraría usted que esto le aportaría beneficios a su organización?* En la Figura 3-63 se puede observar que la funcionalidad de auditoría incluida dentro del modelo apoya a las organizaciones en un 80% arrojando mejores beneficios en su seguimiento y monitoreo a sus empleados.

Figura 3-63. Apoyo a seguimiento y auditoría de los usuarios en la aplicación de negocio.



Fuente: Propia

Al realizar el análisis de los resultados de las entrevistas realizadas para validar el modelo centralizado de control de acceso para las aplicaciones desarrolladas, se logró identificar que son varios los beneficios que se obtienen al disponer de un mecanismo centralizado donde consolide todos los permisos de los usuarios y desde allí llevar a cabo tareas como la asignación o retiro automático, identificar un usuario donde está matriculado y cuáles son sus permisos asignados, disponer de una auditoría donde se indique cuándo y quién otorgó o retiro ese permiso. Esto permite tener una mejor visibilidad de los usuarios que acceden a la información y dependiendo de su contexto o situación actual otorgar o no el acceso a este. En general, consideran que este modelo puede ser implementado como un control de seguridad donde no solo se benefician los analistas de las aplicaciones sino otros actores como el personal de auditoría y los analistas encargados de la ciberseguridad de la información. Con estos resultados arrojados por las entrevistas se puede dar por cumplido el objetivo 4 de este proyecto.

3.5. Resultado consolidado

En el objetivo 1 se logró Identificar cómo manejan la autorización en las aplicaciones a la medida y su gobierno, en al menos once organizaciones; se encontró que tienen varios mecanismos de control de acceso y estos no se integran o comunican para facilitar su administración y auditoría y que consideran que una solución centralizada apoyaría dicha labor. Cada aplicación tiene su propia definición sin ir alineada a la estrategia de la organización, reforzando el planteamiento del problema definido para este proyecto de grado y así mismo, entregando insumos relevantes para el objetivo 3 que es la definición de ese modelo centralizado de autorización.

En el caso del objetivo 2 se logró determinar cuáles son las mejores prácticas para el manejo de la autorización en las organizaciones, se encontraron algunos modelos como DAC, RBAC, ABAC, PBAC, CapBAC, CBAC y Blockchain, y se hizo una comparación entre ellos; sin embargo, por sí solos, no suplían todas las necesidades identificadas. Así mismo, se investigó sobre algunos estándares recomendados de seguridad para el manejo de control de acceso como lo son SAML, OAUTH, XACML, JSON y LDAP, donde, los más recomendados a utilizar son SAML y OAUTH. Dependiendo del tipo de la aplicación y su contexto se utilizaría uno u el otro.

En el objetivo 3 se establecieron los requerimientos mínimos requeridos para el control de acceso en las aplicaciones a la medida, en cuanto a la autorización. Los insumos para la definición de estos requerimientos fueron los objetivos 1 y 2. Se tomó la decisión de crear una sinergia entre varios modelos de control de acceso y así, construir el modelo centralizado de autorización utilizando el control de acceso basado en roles (RBAC), control de acceso basado en atributos (ABAC) y el control de acceso basado en el contexto del usuario (CBAC), de tal forma que permita tanto estática como dinámicamente evaluar el entorno del usuario y determinar si se otorga o no el acceso al recurso solicitado; la interacción entre las aplicaciones de negocio y el componente de autorización se debe realizar utilizando Oauth y consumiendo las APIs expuestas para obtener la configuración del usuario y la aplicación.

Para el objetivo 4 se validó el modelo de seguridad a través de una prueba de concepto, que permitió establecer el nivel de cumplimiento en el control de acceso. Se construyó un prototipo a

través del desarrollo de una aplicación web con las mínimas historias de usuario requeridas para ser validado por algunos de los entrevistados en el objetivo 1. Los entrevistados coincidieron en que son mayores los beneficios al tener centralizado el control de acceso, como se hace con la identidad del usuario utilizando el directorio activo, ya que esto ayuda a la seguridad de la información en las aplicaciones, a disminuir riesgos como fuga de información, al seguimiento de las responsabilidades asignadas a los usuarios en las aplicaciones, y a la toma de acciones inmediatas al estar sincronizada la situación del usuario con el directorio activo. Algunos manifestaban que no solo se debería hacer para las aplicaciones desarrolladas a la medida sino también incluir las que son paquetes comprados (aquellas a las cuales no se tiene acceso al código fuente) para tener un panorama más amplio de todos los permisos que el usuario tiene a cargo.

Al consolidar los resultados obtenidos en estos 4 objetivos se logró cumplir el objetivo general que corresponde a: "Diseñar un modelo centralizado de autorizaciones para mejorar la seguridad en aplicaciones desarrolladas a la medida en las organizaciones con el fin de disminuir malas prácticas de desarrollo en el control de acceso, el robo de información, el desaprovisionamiento inoportuno y la falta de seguimiento y monitoreo".

4. Conclusiones y recomendaciones

Conclusiones

- A partir de los hallazgos encontrados en el objetivo 1: "Identificar cómo manejan la autorización en las aplicaciones a la medida y su gobierno, en al menos cuatro organizaciones", que consistió en realizar encuestas a varias empresas sobre el estado del control de acceso en las aplicaciones desarrolladas a la medida, se sustentó el problema definido para este proyecto de grado donde se identificaron las principales falencias en la asignación o retiro automático de permisos a nivel empresarial, no se tiene un modelo centralizado de control de acceso, el seguimiento o auditoría al control de acceso es débil, no hay un gobierno claro sobre el uso del control de acceso, y con estás debilidades mencionadas previamente se está generando una puerta de entrada a un atacante para modificar o robar información afectando la reputación de la organización.

- Para dar cumplimiento al objetivo 2: "Determinar cuáles son las mejores prácticas para el manejo de la autorización en las organizaciones", se realizaron varias búsquedas en fuentes de información confiables como NIST, IEEE, SANS, Gartner, OWASP, allí se encontraron varios modelos de autorización para el manejo de control de acceso en las aplicaciones desarrolladas, algunos son tradicionales o tienen muchos años en el mercado, otros son nuevos, que surgen con la evolución de la tecnología como es el caso de Blockchain; todo este abanico de opciones permitió definir el modelo requerido para este proyecto de grado y que realmente aporte a las organizaciones.

- En el objetivo 3: " Establecer los requerimientos mínimos requeridos para el control de acceso en las aplicaciones a la medida, en cuanto a la autorización" se logró definir los requisitos que debe cumplir el modelo de autorización centralizado para las aplicaciones desarrolladas a la medida, haciéndola una solución flexible, fácil de implementar y personalizar. Así mismo, se planteó el diseño del modelo utilizando varios diagramas para su entendimiento tanto del cliente como del desarrollador. Con esto se logra minimizar las brechas de seguridad que tienen las aplicaciones desarrolladas a la medida porque existe una base de datos centralizada que está integrada al componente de autenticación para conocer el estado actual de un usuario. Al

momento de cambiar alguna propiedad del usuario estos cambios se van a ver reflejados en las aplicaciones donde se encuentra matriculado, reduciendo la modificación o fuga de información.

- De acuerdo con los resultados obtenidos para el objetivo 4: " Validar el modelo de seguridad a través de una prueba de concepto, que permita establecer el nivel de cumplimiento en el control de acceso " se puede concluir que el modelo centralizado de autorización ayuda a disminuir malas prácticas de desarrollo en el control de acceso debido a que todos los desarrolladores deben seguir un mismo lineamiento definido a nivel organizacional; evitar el robo de información porque dinámicamente se valida el contexto y situación del usuario, en caso de estar inactivo o encontrarse con alguna restricción de horario de uso, éste no podrá acceder al recurso solicitado; mejorar el desaprovisionamiento inoportuno porque al integrarse con el directorio activo y cambiar el estado del usuario a retirado, la sincronización con el componente de autorización será automática y sus permisos serán revocados; tener reportes sobre los accesos otorgados a los usuarios y conocer donde se encuentra matriculado este. Los entrevistados quedaron satisfechos con el modelo presentado y considerar que aportaría gran valor en las organizaciones.

- El modelo propuesto permite integrar todas las aplicaciones desarrolladas a la medida en un único componente centralizado para el control de acceso que permita, de manera eficiente y oportuna, realizar algún cambio en el permiso del usuario y este se refleje en todas las aplicaciones en las que tiene acceso, preservando así el principio de la confidencialidad de la información en los sistemas y reduciendo los riesgos de seguridad, cumpliendo así con el objetivo general de este proyecto.

Recomendaciones

- Extender el modelo hacia otro tipo de aplicaciones como son los dispositivos móviles, incluso hacer integración con las aplicaciones que son compradas para tener una mayor cobertura de los permisos otorgados a los usuarios y poder monitorear su comportamiento.
- Como trabajo futuro se puede plantear el modelo de control de acceso utilizando tecnología Blockchain para identificar cuál solución es más eficiente y rentable para las organizaciones.
- Con el fin de facilitarle al desarrollador la implementación del control de acceso en las aplicaciones de negocio desarrollas a la medida, construir en los entornos de desarrollo más reconocidos como Visual Studio, Android, Eclipse un plugin o extensión que permita instalarse en la aplicación y facilitar la integración y desarrollo con el componente de autorización.
- El modelo actual solo incluye la conexión a un solo componente de autenticación, una mejora a éste es diseñar cómo sería la integración con varios proveedores de identidad que permita la validación de los usuarios en diferentes repositorios.

Anexos

Anexo A: Encuesta de autorización de aplicaciones

Formato Encuesta Autorizacion App_v4.c

Anexo B: Entrevista para la validación del modelo centralizado

Entrevista validación Modelo.docx

Bibliografía

[1] E. Del Peso Navarro and M. A. Ramos González, “La seguridad de los datos de carácter personal,” in *La seguridad de los datos de caracter personal*, 2nd ed., D. de Santos, Ed. 2015, p. 220.

[2] M. Á. Caballero Velasco and D. Cilleros Serrano, *Ciberseguridad y transformación digital*. 2019.

[3] P. Łąka, “The risks of not having an Identity and Access Management system,” 2019. [Online]. Available: https://www.e-point.com/blog/the-risks-of-not-having-an-identity-and-access-management-system. [Accessed: 18-Nov-2019].

[4] ICONTEC, *Norma Técnica NTC / ISO / IEC 27002. Tecnología de la Información. Técnicas de seguridad. Códido de práctica para controles de seguridad de la información*. 2015.

[5] C. Gutiérrez, “Lo que muchos saben pero pocos aplican sobre desarrollo seguro | WeLiveSecurity,” 2014. [Online]. Available: https://www.welivesecurity.com/la-es/2014/02/03/muchos-saben-pero-pocos-aplican-sobre-desarrollo-seguro/. [Accessed: 17-Nov-2019].

[6] Contrast Security, “CONTRAST LABS APPLICATION SECURITY INTELLIGENCE BIMONTHLY REPORT,” 2020.

[7] ElevenPaths, “Informe sobre el estado de la seguridad 2019 H1,” pp. 13–15, 2019.

[8] Positive Technologies, “Web application vulnerabilities and threats: statistics for 2019,” 2019. [Online]. Available: https://www.ptsecurity.com/upload/corporate/ww-en/analytics/web-vulnerabilities-2020-eng.pdf. [Accessed: 07-Sep-2020].

[9] F. Gaehtgens, K. Kampman, and B. Iverson, “Magic Quadrant for Identity Governance and Administration ID: G00326925,” 2018.

[10] H. Teixeira and K. Kampman, “IGA Best Practices: Prioritize Analytics When Adopting IGA ID: G00388046,” 2019.

[11] ICONTEC, *Norma Técnica NTC/ISO/IEC 27000. Tecnología de información. Técnicas de seguridad. Sistemas de gestión de seguridad de la información (SGSI). Visión general y vocubulario*. 2017.

[12] ISACA, “ISACA Interactive Glossary & Term Translations | ISACA.” pp. 1–72, 2015.

[13] Gartner Inc., “Authorization - Gartner IT Glossary,” *Gartner*, 2019. [Online]. Available:

https://www.gartner.com/it-glossary/authorization. [Accessed: 24-May-2019].

[14] V. C. Hu, R. Chandramouli, D. F. Ferraiolo, and D. R. Kuhn, *Attribute-Based Access Control*. Norwood, MA, Estados Unidos, 2017.

[15] D. Kononov and S. Isaev, "Improving Web Applications Security Using Path-Based Role Access Control Model," in *2018 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC)*, 2018, pp. 1–4.

[16] OWASP, "Access Control - OWASP," 2016. [Online]. Available: https://owasp.org/www-community/Access_Control. [Accessed: 25-Aug-2019].

[17] V. C. Hu *et al.*, "NIST Special Publication 800-162 Guide to Attribute Based Access Control (ABAC) Definition and Considerations," 2019.

[18] V. C. Hu, D. F. Ferraiolo, and D. R. Kuhn, "NIST SP 800-205, Attribute Considerations for Access Control Systems," *NIST*, 2019.

[19] Farahmand Homan, "A Systematic and Practical Approach to Optimizing Authorization Architecture - ID: G00291477," *Gartner*, 2015.

[20] R. Rojas, *Los ordenadores de Konrad Zuse*, vol. 255. Dialnet. Investigación y ciencia, 1997.

[21] R. Macau, "TIC: ¿PARA QUÉ? (Funciones de las tecnologías de la información y la comunicación en las organizaciones)," *Revista de Universidad y Sociedad del Conocimiento, vol. I No. 1*, p. 12, 2004.

[22] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Computer role-based access control models," *IEEE*, vol. 29, no. 2, pp. 38–47, Feb-1996.

[23] K. Schwaber and J. Sutherland, *La Guía de Scrum*. 2020.

[24] J. Abad and L. Salazar, *Historias de usuario: Una visión pragmática*. Independently Published, 2018.

[25] K. Beck *et al.*, "Principios del Manifiesto Ágil." [Online]. Available: http://agilemanifesto.org/iso/es/principles.html. [Accessed: 24-Feb-2021].

[26] M. A. Ortega and E. D. Camacho, "Uso de los modelos tradicionales y las metodologías ágiles aplicadas en la industria de software colombiano," Cali, 2019.

[27] MITRE, "Acerca de CWE," 2020. [Online]. Available: https://cwe.mitre.org/about/index.html. [Accessed: 26-Apr-2020].

[28] Mitre, "CWE-264: Permissions, Privileges, and Access Controls (3.3)," 2019. [Online]. Available: https://cwe.mitre.org/data/definitions/264.html. [Accessed: 25-Aug-2019].

[29] Mitre, “CWE-284: Improper Access Control,” 2019. [Online]. Available: https://cwe.mitre.org/data/definitions/284.html. [Accessed: 25-Aug-2019].

[30] MITRE, “CWE-285: Autorización incorrecta,” 2020. [Online]. Available: https://cwe.mitre.org/data/definitions/285.html. [Accessed: 26-Apr-2020].

[31] MITRE, “Vulnerability distribution of cve security vulnerabilities by types,” *MITRE*, 2019. [Online]. Available: https://www.cvedetails.com/vulnerabilities-by-types.php. [Accessed: 20-Apr-2020].

[32] K. Watson, “DECEMBER 2019 AppSec Intelligence Report,” 2020. [Online]. Available: https://www.contrastsecurity.com/security-influencers/december-2019-appsec-intelligence-report. [Accessed: 07-Mar-2020].

[33] EY, “¿La ciberseguridad es algo más que protección?,” *EY*, p. 80, 2018.

[34] Ruddy Mary, “Key Features for Customer Identity and Access Management ID: G00378004,” *Gartner*, 2019.

[35] Kampman Kevin, “Why You Need an IAM Program ID: G00319774,” 2017.

[36] SailPoint, “IAM - Gestión de accesos e identidades | SailPoint Technologies,” 2019. [Online]. Available: https://www.sailpoint.com/es/identity-library/identity-and-access-management/. [Accessed: 24-May-2019].

[37] National Institute of Standards and Technology (NIST), “SP 800-53 Rev.4 - Security and Privacy Controls for Federal Information Systems and Organizations,” *Natl. Inst. Stand. Technol. - Spec. Publ.*, p. 462, Jan. 2015.

[38] OWASP, “OWASP Top 10 - 2017.” p. 25, 2017.

[39] D. Arias, “Facebook estaría usando números de teléfono sin autorización • ENTER.CO,” *ENTER.CO*, 2019. [Online]. Available: https://www.enter.co/cultura-digital/redes-sociales/facebook-numeros-telefono/. [Accessed: 25-Aug-2019].

[40] Gartner, “Gartner Forecasts Worldwide Information Security Spending to Exceed $124 Billion in 2019,” 2018. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2018-08-15-gartner-forecasts-worldwide-information-security-spending-to-exceed-124-billion-in-2019. [Accessed: 14-May-2019].

[41] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, “Smart contract-based access control for the internet of things,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1594–1605, Apr. 2019.

[42] I. Indu and P. M. Rubesh Anand, “Hybrid Authentication and Authorization Model for Web based Applications,” *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, p. 5, 2016.

[43] A. Sergeev and R. Matulevicius, “An Approach to Capture Role-Based Access Control Models from Spring Web Applications,” in *Proceedings - 2017 IEEE 21st International Enterprise Distributed Object Computing Conference, EDOC 2017*, 2017, vol. 2017-January, pp. 159–164.

[44] E. Rissanen, “xacml-3.0-core-spec-os-en eXtensible Access Control Markup Language (XACML) Version 3.0 OASIS Standard Specification URIs,” 2013.

[45] M. Uddin and S. Islam, “A dynamic access control model using authorising workflow and task-role based access control,” *IEEE Access*, pp. 1–1, 2019.

[46] R. Pressman, *Ingeniería del software. Un enfoque práctico.*, 7th ed. McGraw-Hill Interamericana, 2010.

[47] OWASP, “Access Control Cheat Sheet.” [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Access_Control_Cheat_Sheet.html. [Accessed: 04-Jul-2020].

[48] O. Gómez Baryolo, “MODELO DE CONTROL DE ACCESO PARA SISTEMAS DE INFORMACIÓN BASADOS EN TECNOLOGÍAS WEB,” *Revista Científica ECOCIENCIA, Diciembre, Vol.5, Número 6*, Ecuador, p. 33, Dec-2018.

[49] S. Gusmeroli, S. Piccione, and D. Rotondi, “A capability-based security approach to manage access control in the Internet of Things,” *Mathematical and Computer Modelling*, vol. 58, no. 5–6, Pergamon, pp. 1189–1205, Sep-2013.

[50] B. Shebaro, O. Oluwatimi, and E. Bertino, “Context-based access control systems for mobile devices,” *IEEE Trans. Dependable Secur. Comput.*, vol. 12, no. 2, pp. 150–163, Mar. 2015.

[51] A. Salas Vázquez, “Episodio clínico en blockchain de Ethereum,” Jan. 2020.

[52] M. U. Rahman, “Scalable Role-based Access Control Using The EOS Blockchain,” Jul. 2020.

[53] D. Arroyo Guardeño, J. Díaz Vico, and L. Hernández Encinas, *Blockchain*. LA CATARATA, 2019.

[54] P. Madsen and E. Maler, “SAML V2.0 Executive Overview,” 2005.

[55] B. Cook and C. Messina, “OAuth 2.0.” [Online]. Available: https://oauth.net/2/. [Accessed: 09-Nov-2019].

[56] D. Hardt, “RFC 6749 - The OAuth 2.0 Authorization Framework.” Internet Engineering Task

Force (IETF), Wilmington, USA, 2012.

[57] OASIS, “A Brief Introduction to XACML,” 2003. [Online]. Available: https://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html. [Accessed: 03-May-2020].

[58] Auth0, “JSON Web Token Introduction - jwt.io.” [Online]. Available: https://jwt.io/introduction/. [Accessed: 15-May-2020].

[59] M. B. Jones, J. Bradley, and N. Sakimura, “RFC 7519 - JSON Web Token (JWT),” May-2015. [Online]. Available: https://tools.ietf.org/html/rfc7519. [Accessed: 15-May-2020].

[60] J. Sermersheim, “RFC 4511 - Lightweight Directory Access Protocol (LDAP): The Protocol,” *IETF*, Jun-2006. [Online]. Available: https://tools.ietf.org/html/rfc4511. [Accessed: 05-Jul-2020].

[61] K. D. Zeilenga, “RFC 4510 - Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map,” *IEFT*, Jun-2006. [Online]. Available: https://tools.ietf.org/html/rfc4510. [Accessed: 15-May-2020].

[62] J. T. Mejía Viteri, M. I. Gonzáles Valero, and Á. R. España León, “Gestión de Usuarios Con LDAP (Lightweight Directory Access Protocol) para el Acceso a los Servicios Tecnológicos y a la Información en las Empresas,” *Revista ciencia e investigación*, Aug-2016. [Online]. Available: https://revistas.utb.edu.ec/index.php/sr/article/view/84/66. [Accessed: 05-Jul-2020].

[63] OWASP, “OWASP Application Security Verification Standard,” *OWASP*, Mar-2019. [Online]. Available: https://owasp.org/www-project-application-security-verification-standard/. [Accessed: 20-Aug-2020].

[64] F. Gaehtgens, K. Kampman, A. Data, H. Teixeira, and D. Collinson, “Magic Quadrant for Identity Governance and Administration,” Oct. 2019.