Universidad de Matanzas
Facultad de Ciencias Técnicas
Departamento de Informática

TRABAJO DE DIPLOMA EN OPCIÓN DEL TÍTULO DE INGENIERO INFORMÁTICO

SISTEMA DE GESTION DE ROLES

Autor: Jahaziel Betancourt Rodríguez.
Tutor: Dr.C. Josue Segura Montero.

Matanzas, 2023

Resumen

La aplicación Web fue realizada en la Empresa de Tecnologías de la Información para la Defensa (XETID). La propuesta se fundamenta en desarrollar un *Sistema para la Gestión de Roles*, que permita gestionar el control de acceso de usuarios de las aplicaciones informáticas basado en un componente que se ajuste a las técnicas de control de acceso basado en roles (RBAC). Entre las facilidades que brinda el sistema se encuentran gestionar de forma integrada y segura la información de módulos y funcionalidades, y los roles del negocio de cada una de las aplicaciones, así como la gestión de acceso de los usuarios mediante la asignación de roles y entidades. El sistema posibilita gestionar la información de las aplicaciones de los módulos o subsistemas que componen una aplicación informática El sistema facilita la gestión de perfiles de usuario de aplicaciones que se ha registrados en las aplicaciones previamente, modificar el acceso a otras aplicaciones y cambiar el rol y estructura para cada una de ellas. Se empleó la metodología de desarrollo de software Proceso de Desarrollo de Software (ProDeSoft 2012) , el lenguaje de programación PHP 8 y como gestor de base de datos PgAdmin. Se obtuvo un sistema funcional, el cual se le realizaron una serie de pruebas que permitieron constatar su funcionalidad y verificar que cumpliera con las necesidades y exigencias del cliente.

Abstract

The Web application was made at the Defense Information Technology Company (XETID). The proposal is based on developing a System for Role Management, which allows managing user access control of computer applications based on a component that adjusts to role-based access control (RBAC) techniques. Among the facilities provided by the system are the integrated and secure management of module and functional information, the business roles of each of the applications, as well as user access management through the assignment of roles and entities. . The system makes it possible to manage the application information of the modules or subsystems that make up a computer application. The system makes it easy to manage user profiles of previously registered applications, modify access to other applications and change the role and structure for each of them. The software development methodology was used Software Development Process (ProDeSoft 2012), the PHP 8 programming language and PgAdmin as a database manager. A functional system was obtained, which underwent a series of tests that allowed us to verify its functionality and verify that it met the needs and demands of the client.

Contenido

Introducción

La seguridad de la información es un concepto que se involucra cada vez más en muchos aspectos de la sociedad interconectada, en gran parte como resultado de la adopción casi ubicua de las tecnologías de la información y comunicación.
Según (ISO/IEC 27000 2018) la seguridad de la información se podría definir como aquellos procesos, buenas prácticas y metodologías que busquen proteger la información y los sistemas de información del acceso, uso, divulgación, interrupción, modificación o destrucción no autorizada. Esta definición básicamente significa que se debe proteger los datos y recursos de infraestructura tecnológica de aquellos quiénes intentarían hacer un mal uso de ellos.
La seguridad de la información, como concepto, se basa en cuatro pilares: la disponibilidad: acceso a la información cuando se requiere, teniendo en cuenta la privacidad, integridad: información correcta sin modificaciones no autorizadas ni errores, confidencialidad: información accesible solo para personal autorizado, integridad: información correcta sin modificaciones no autorizadas ni errores, autenticación: Información procedente de un usuario que es quien dice ser. Se verifica y se debe garantizar que el origen de los datos es correcto(Vega 2021)
Conforme crece la utilización de las TIC crece la necesidad de ofrecer a las organizaciones mecanismos para una mejor implementación de la seguridad de la información. Dichos mecanismos deben permitir controlar el acceso a los recursos, gestionar usuarios y sus datos de identificación, asociar roles, perfiles y políticas de seguridad.
Estos suelen estar formados por dispositivos de autenticación que facilitan el control del acceso lógico de los usuarios en los sistemas informáticos debido a que la ausencia de seguridad pone en riesgo la información y conlleva a considerables pérdidas monetarias.
Uno de los conceptos fundamentales asociados con la seguridad de la información precisamente está asociado al control de acceso. En este campo los países desarrollados como Estados Unidos, Francia, Inglaterra y China se destacan en la producción y utilización de controladores de acceso automatizados.
En el ámbito internacional, (Wang, et al. 2014) propusieron un modelo extendido de control de acceso basado en roles para un sistema empresarial de gestión del conocimiento.(Yang et al. 2013) propusieron un modelo de control de acceso basado en diferentes tipos de usuarios o roles para una plataforma de trabajo colaborativo. (Tounis, Kifayat y Merabti 2014) describen que el modelo de control de acceso basado en roles, es considerado como una forma natural de controlar el acceso a los recursos en las organizaciones y empresas.(Rivas 2016) propone la implementación de un sistema de control de acceso que pueda ser integrado al sistema de servidores de archivos donde es almacenada la información de la empresa SNX S.A.C.
En el ámbito nacional se han desarrollado aplicaciones para el control de acceso a aplicaciones informáticas, a saber: Componente para facilitar el proceso de autenticación de usuarios en aplicaciones informáticas en instituciones de salud (Téllez y Guevara 2015), sistema de control de acceso a los laboratorios de producción (UCILAB)(Pérez 2016) y sistema para el control de acceso a los

laboratorios del Centro de Tecnologías para la Formación desarrollo de un sistema para el control de acceso a los laboratorios del CentroFORTES (Freixas y Noa 2015)

Entre los tipos de control de acceso se encuentra, el control de acceso basado en roles (RBAC) el cual es una función de seguridad para controlar lo que los usuarios pueden hacer dentro de los sistemas de TI de una empresa. Consiste en asignar a cada usuario uno o varios "roles" y al otorgar a cada rol diferentes permisos. El RBAC puede aplicarse a una sola aplicación de software o a varias aplicaciones (Cloudflare Connect 2023)

Los procesos de autenticación y verificación de usuarios de las aplicaciones informáticas desarrolladas por la Empresa de Tecnologías de la Información para la Defensa (XETID), generalmente, se gestionan de manera individual, por lo que cada usuario tiene que emplear varias cuentas para poder hacer uso de todas las aplicaciones que se ponen a su disposición por parte de la empresa, por tanto, la entidad se ha planteado como propósito desarrollar e implantar una herramienta que gestione las cuentas de usuarios y pueda brindar esta información de manera segura e integrada.

Partiendo de estas necesidades y sobre la base de la situación problemática planteada, se formula el siguiente problema científico: ¿Cómo contribuir a la gestión de roles de las aplicaciones informáticas la Empresa XETID mediante su automatización?

Por lo que se plantea la siguiente **hipótesis**: si se desarrolla un *Sistema para la Gestión de Roles*, entonces se puede gestionar el control de acceso de usuarios de las aplicaciones informáticas de forma integrada y más segura.

La Gestión de Roles constituye el **objeto de estudio** de esta investigación. Se define como campo de acción la automatización de la gestión de roles de las aplicaciones informáticas de la Empresa XETID.

El objetivo general de la misma es desarrollar un sistema Informático para la gestión de roles de las aplicaciones informáticas de la Empresa XETID.

Para cumplir el objetivo general, se desarrollan los siguientes objetivos específicos:

- Determinar el estado del arte de la temática y las tendencias tecnológicas actuales necesarias para el desarrollo de la propuesta de solución.
- Diseñar la propuesta de solución a través de la metodología de software seleccionada.
- Validar la propuesta de solución mediante las pruebas.

Variable independiente: sistema Informático para la gestión de roles de las aplicaciones informáticas de la Empresa XETID.

Variable dependiente: gestionar el control de acceso de usuarios de las aplicaciones informáticas de forma integrada y más segura.

Para darle solución al problema que se plantea y cumplir los objetivos propuestos se utilizaron los métodos siguientes:

Teóricos

- Histórico-Lógico: Se utilizó durante la revisión de la literatura existente sobre el tema para analizar los antecedentes y evolución de este tipo de sistemas informáticos tanto dentro como fuera del país.
- Análisis y síntesis: Se consultó literatura especializada y documentos rectores de la teoría que se utiliza en la gestión de roles.
- Hipotético- deductivo: Para el análisis y formulación de la hipótesis.
- Modelación: Para modelar los procesos de negocio.

Empíricos

- Entrevistas: Fue un método que se utilizó con bastante frecuencia para levantar los requisitos y dar los primeros pasos en el entendimiento del problema.
- Entre los **aportes** de la investigación se destacan:
- El teórico-investigativo, al integrar los procedimientos tradicionales más utilizados por autores relacionados con el tema, a través de los diferentes artefactos de la metodología de desarrollo de software que permitió orientar metodológicamente la secuencia de acciones lógicas a desarrollar; y los elementos a tener en cuenta para la continuidad de la investigación,
- El práctico, al desarrollar una herramienta automatizada que permita gestionar el control de acceso de usuarios de las aplicaciones informáticas de forma integrada y más segura en la Empresa XETID.
- Entre los resultados esperados con esta investigación, se encuentran: contar con una herramienta que permita gestionar de forma integrada y segura la información de módulos y funcionalidades, y los roles del negocio de cada una de las aplicaciones, así como la gestión de acceso de los usuarios mediante la asignación de roles y entidades.
- Atendiendo a lo planteado, la investigación queda estructurada en introducción, tres capítulos, conclusiones, recomendaciones y referencia bibliográficas según se observa:
- Introducción: Se caracteriza la situación problemática y se fundamenta el problema científico a resolver.
- Capítulo I: Marco teórico-referencial del tema a tratar, donde se plantean los conceptos fundamentales asociados con el tema. Además, se realiza un análisis de las tendencias tecnológicas que serán utilizadas, así como la metodología desarrollo herramientas y tecnologías.
- Capítulo II: Diseño y construcción de la Solución Propuesta, donde se argumenta la solución propuesta al problema y se describe la implementación del software a través de la metodología Prodesoft.
- Capítulo III: Validación de la solución propuesta, donde se muestran las principales interfaces del prototipo inicial, y se realiza un análisis de los resultados obtenidos.
- Un apartado de conclusiones donde se verifica el cumplimiento de los objetivos trazados al inicio de la investigación.

- Las recomendaciones en la cual se plasman una serie de propuestas encaminadas a la continuidad de esta investigación.
- Y las referencias de la bibliografía citada.

Capítulo 1

Para realizar una buena investigación siempre es necesario tener bien claro cuáles son las bases que la sustentan y le dan firmeza, por eso este capítulo estará dedicado a conocer mejor los detalles del sistema que se pretende realizar, partiendo de un estudio de sus antecedentes. Además, se mostrarán las tecnologías y herramientas que se utilizarán para el desarrollo del mismo, así como un acercamiento a la tecnología utilizada.

1.1 Descripción de la entidad

La Empresa de Tecnologías de la Información para la Defensa (XETID), empresa cubana líder en el desarrollo de soluciones tecnológicas para la informatización de la sociedad, asume como misión la proyección, diseño, desarrollo y comercialización de productos y servicios, a partir del uso de las tecnologías de la informática y las comunicaciones.

Actualmente el desarrollador de la XETID tiene que controlar de manera manuscrita la información específica y perfiles de usuarios de cada una de las aplicaciones que desarrolla. Cómo parte de la estrategia de desarrollo basado en componentes, en aras de potenciar la reutilización y minimizar los tiempos de desarrollos se decide por parte de la empresa centralizar los procesos de autenticación, autorización y verificación en un componente que se ajuste a las técnicas de control de acceso basado en roles (RBAC).

1.2 Definición de Control de acceso

El control de acceso es un elemento esencial de seguridad que determina quién puede acceder a ciertos datos, aplicaciones y recursos, y en qué circunstancias. De la misma forma que las claves y listas de invitados con aprobación previa protegen los espacios físicos, las directivas de control de acceso protegen los espacios digitales. En otras palabras, permiten a las personas adecuadas entrar y mantener a las personas equivocadas fuera. Las directivas de control de acceso se basan en gran medida en técnicas como autenticación y autorización, que permiten a las organizaciones comprobar explícitamente que los usuarios son quienes dicen ser y que a estos usuarios se les concede el nivel adecuado de acceso en función del contexto, como el dispositivo, la ubicación, el rol y mucho más.(Microsoft 2023b)

El control de acceso evita que la información confidencial, como los datos de los clientes y la propiedad intelectual, sea sustraída por usuarios no autorizados. También reduce el riesgo de filtrado de datos por parte de los empleados y mantiene a raya las amenazas web. En lugar de gestionar los permisos manualmente, la mayoría de las organizaciones impulsadas por la seguridad recurren a soluciones de gestión de identidades y accesos para implementar directivas de control de acceso(Microsoft 2023b)

1.3 **Tipo de Controles de acceso**

1.3.1 Control de acceso obligatorio (MAC)

En el modelo de control de acceso obligatorio (Mandatory Access Control, MAC) todos los sujetos y objetos son clasificados basándose en niveles predefinidos de seguridad que son usados en el proceso de obtención de los permisos de acceso. Para describir estos niveles de seguridad todos los sujetos y objetos son marcados con etiquetas de seguridad que siguen el modelo de clasificación de la información militar (desde "desclasificado" hasta "alto secreto"), formando lo que se conoce como política de seguridad multinivel. Los modelos MAC proporcionan mecanismos sólidos para la protección de datos y tratan con requerimientos de seguridad específicos, así como, los requerimientos derivados de las políticas de control de los flujos de información. Además, es el sistema quien protege los recursos u objetos y el administrador es el que impone las reglas de forma segura. Sin embargo, asegurar las políticas MAC es a menudo una tarea difícil dado que no posee suficiente flexibilidad (MAC 2023)

1.3.2 El modelo de control de acceso discrecional.

(Discretionary Access Control, DAC), también llamado modelo de seguridad limitada, es un modelo no orientado al control del flujo de información. Todos los sujetos y objetos en el sistema son controlados y se especifican reglas de autorización de acceso para cada sujeto y objeto. Los sujetos pueden ser usuarios, grupos o procesos. Los modelos DAC están basados en la idea de que el propietario de un objeto, su autor, tiene el control sobre los permisos del objeto, es decir, el autor es autorizado a permitir u otorgar permisos para este objeto a otros usuarios (KeepCoding 2023)

1.3.3 Control de Acceso Basado en Roles (RBAC)

En 1992 surgió un nuevo modelo independiente de los otros dos, control de acceso basado en roles (Role-Based Access Control, RBAC). RBAC está basado en la definición de un conjunto de elementos y las relaciones entre ellos. A nivel general describe un grupo de usuarios que pueden estar actuando bajo un conjunto de roles y realizando operaciones en las que utilizan un conjunto de recursos. En una organización, un rol puede ser definido como una función que describe la autoridad y responsabilidad dada a un usuario en un instante determinado. Incluye un conjunto de sesiones que constituyen la relación entre un usuario y un subconjunto de roles que son activados en el momento de establecer dicha sesión. Cada sesión está asociada con un único usuario, un usuario puede tener una o más sesiones asociadas.

Los permisos disponibles para un usuario son el conjunto de permisos asignados a los roles que están activados en todas las sesiones del usuario, sin tener en cuenta las sesiones establecidas por otros usuarios en el sistema. Añade la posibilidad de modelar una jerarquía de roles de forma que se puedan realizar generalizaciones y especializaciones en los controles de acceso y se facilite la modelización de la seguridad en sistemas complejos (BAC 2023)

En los modelos RBAC, los derechos de acceso se conceden de acuerdo con funciones empresariales definidas, en lugar de basarse en la identidad o

antigüedad de las personas. El objetivo es proporcionar a los usuarios únicamente los datos que necesitan para realizar su trabajo y nada más. (Microsoft, 2023)

El RBAC es una función de seguridad para controlar lo que los usuarios pueden hacer dentro de los sistemas de TI de una empresa. Consiste en asignar a cada usuario uno o varios "roles" y al otorgar a cada rol diferentes permisos. El RBAC puede aplicarse a una sola aplicación de software o a varias aplicaciones. (Cloudflare, 2023).

La propuesta de solución en la investigación se ajusta a las técnicas de control de acceso basado en roles (RBAC) con el objetivo de centralizar los procesos de autenticación, autorización y verificación de las aplicaciones entidad en aras de una mayor seguridad e integración.

1.4 Antecedentes

1.4.1 (Wang, et al. 2014), propusieron un modelo extendido de control de acceso basado en roles para un sistema empresarial de gestión del conocimiento. Al modelo de control de acceso basado en roles, los autores incluyen una estructura de grupos de usuarios, la cual a su vez era dividida en dos tipos de grupos, esto con el fin de que los usuarios puedan ser agregados a los dos tipos diferentes, y los privilegios puedan ser otorgados tanto a los grupos de usuarios como a los roles.

1.4.2 (Yang et al. 2013), propusieron un modelo de control de acceso basado en diferentes tipos de usuarios o roles para una plataforma de trabajo colaborativo. El modelo de control de acceso para la plataforma de trabajo colaborativo estuvo definido por tres elementos principalmente: un conjunto de usuarios (compuesto por los usuarios del sistema o cualquier otra entidad que actué como un usuario), un conjunto de objetos (compuesto por los recursos del sistema) y un conjunto de operaciones (compuesto por entidades que definen que es lo que se debe hacer).

1.4.3 (Rivas 2016) propone la implementación de un sistema de control de acceso que pueda ser integrado al sistema de servidores de archivos donde es almacenada la información de la empresa SNX S.A.C y de esa manera poder mejorar sus procesos de seguridad de la información.

1.4.4 Componente para facilitar el proceso de autenticación de usuarios en aplicaciones informáticas en instituciones de salud.

El propósito de este trabajo es desarrollar un componente que centralice y unifique la autenticación del usuario reduciendo la cantidad de acciones de autenticación en las aplicaciones que el profesional de la salud desee emplear en su estación de trabajo y agilizando la atención al paciente(Téllez y Guevara 2015)

1.4.5 Sistema de control de acceso a los laboratorios de producción (UCILAB)

Este sistema lleva el control de los proyectos que radican en los laboratorios destinados a los procesos productivos y por tanto de las personas que pueden tener acceso a dichos laboratorios. En este sistema se chequea qué personas tienen acceso o no a los laboratorios, verificando que estén en la base de datos correspondiente, mediante el número de la identificación. Existen varias

implementaciones de este sistema en la UCI, cada una de ellas específica para el área productiva donde se encuentra, lo que hace que no exista una base de datos centralizada con todos los datos referentes a todos los laboratorios de producción.(Pérez 2016)

1.4.6 Sistema para el control de acceso a los laboratorios del Centro de Tecnologías para la Formación desarrollo

Sistema para el control de acceso a los laboratorios del Centro de Tecnologías para la Formación desarrollo de un sistema para el control de acceso a los laboratorios del CentroFORTES (SCAFORTES). El sistema permite informatizar el monitoreo que se realiza del personal que acede a los laboratorios y a las estaciones de trabajo, generando además un reporte de los incidentes recogidos en cada estación(Freixas y Noa 2015).

En la investigación realizada, se han analizado que los softwares que existen a nivel internacional no son softwares gratuitos y al mismo tiempo no siguen las legislaciones, regulaciones y normas cubanas por lo que no constituyen un elemento para su consulta. En la revisión de las aplicaciones cubanas se dirigen fundamentalmente al control físico de acceso de los usuarios o no cumplen con los requisitos enumerados por el cliente en relación con la gestión control de acceso basado en roles de las aplicaciones informáticas de una forma segura e integrada.

1.5 Metodología de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. El objetivo de las distintas metodologías es el de intentar organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible. (Santander, 2020)

1.5.1 Proceso de Desarrollo de Software (ProDeSoft) v1.7

Proceso de Desarrollo de Software (ProDeSoft 2012) es la metodología que utiliza la XETID en sus proyectos. Según su documento oficial esta metodología divide el ciclo de vida del producto de la siguiente forma:

Figura 1 Etapas del Ciclo de Vida del Producto. Fuente: (Metodología PDS v1.7 XETID)

Modelación: Se define una visión preliminar de la problemática a resolver por medio de la informatización, los antecedentes de las soluciones informatizadas similares del problema planteado u otras experiencias similares. Se reflejan las generalidades de la propuesta de solución, el tipo de solución a emplear. Además de las principales características técnicas que tendrá la solución definiendo con estos elementos un proyecto técnico. Se capturan las partes esenciales del sistema, donde se identifican los procesos de negocio fundamentales y se aceptan los requerimientos funcionales, obteniéndose la línea base de la arquitectura y una estrategia de construcción de la aplicación aprobada por los implicados en el proyecto. (XETID, 2020)

Construcción: Se aclaran los requisitos restantes y se completa el desarrollo del sistema sobre una base estable de la arquitectura. Las fases anteriores sólo dieron una arquitectura básica que es aquí refinada de manera incremental conforme se construye el producto. En esta fase todas las características, componentes, y requerimientos deben ser integrados, implementados, y probados en su totalidad, obteniendo una versión liberada del producto. (XETID, 2020)

Explotación Experimental: Se convierte la versión liberada del producto en una solución estable, donde se eliminan los errores que surgen durante las pruebas y se obtiene una certificación funcional y de seguridad del producto. Esta fase solo se ejecuta cuando se tiene como cliente al Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR), por requerimientos que exige este órgano. (XETID, 2020)

Despliegue: se instala y configura el sistema para un ambiente de

producción real, se capacita al personal que usará la aplicación y se continúa dando soporte durante la explotación del sistema, culminando de ser preciso con transferencias tecnológicas. (XETID, 2020)

1.6 Tecnologías empleadas

1.6.1 Lenguaje de Modelado:

Lenguaje de Modelado Unificado (UML por sus siglas en inglés). Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.(OMG 2023).Este lenguaje se utilizó para el modelado del proceso.

1.6.2 Herramienta de Modelado:

Visual Paradigm v5: Visual Paradigm es una herramienta que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación(Visual Paradigm 2023)Esta se tomó para realizar todos los diagramas que propone (ProDeSoft 2012)t como parte del modelado del negocio.

1.6.3 IDE de Desarrollo:

PhpStorm: es un entorno de desarrollo integrado (IDE) para desarrolladores de PHP creado para maximizar la productividad de los desarrolladores. La aplicación de escritorio IDE le ayuda a escribir, editar, analizar, refactorizar, probar y depurar código PHP en Windows, macOS y Linux. Con PhpStorm, tiene soporte completo para desarrollar aplicaciones en PHP 5.3 y todas las versiones posteriores de PHP. Además, el IDE tiene soporte integrado para HTML5, CSS, JavaScript y XML. Se puede agregar soporte para otros idiomas mediante complementos(JetBrains 2023).

1.6.4 Lenguaje de Programación:

PHP 8: acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, siendo así sencillo de aprender. El objetivo principal de este lenguaje es permitir a los desarrolladores web escribir dinámica y rápidamente páginas web generadas; aunque se puede hacer mucho más con PHP.

Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno(«PHP» 2023).Este lenguaje se utilizó para la programación de la lógica del negocio en los plugin.

1.6.5 Motor de Base de Datos (BD):

PostgreSQL: es un sistema de gestión de bases de datos relacionales de objetos (ORDBMS) basado en POSTGRES, versión 4.2, desarrollado en el Departamento de Ciencias de la Computación de la Universidad de California en Berkeley. POSTGRES fue pionero en muchos conceptos que sólo estuvieron disponibles en algunos sistemas de bases de datos comerciales mucho más tarde.(PostgreSQL 2023)

1.6.6 Herramienta de Gestión de BD:

PgAdmin 4: es la herramienta líder de gestión de código abierto para Postgres, la base de datos de código abierto más avanzada del mundo. pgAdmin 4 está diseñado para satisfacer las necesidades de usuarios de Postgres tanto principiantes como experimentados, proporcionando una potente interfaz gráfica que simplifica la creación, el mantenimiento y el uso de objetos de bases de datos(pgAdmin 2023) Esta herramienta se empleó como gestor de BD de la aplicación.

1.6.7 Framework de Desarrollo:

Symfony 6: es un popular framework de PHP para desarrollo de aplicaciones web y de soporte, y un conjunto de segmentos de PHP reutilizables que pueden ser fácilmente utilizados por cualquier desarrollador que hacen que el trabajo sea más fluido y fácil.(Symfony 2023)

Doctrine: Es un ORM o (Object Relation Mapper) es una técnica de programación que nos permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de nuestra base de datos pasan a ser clases y los registros objetos que podemos manejar con facilidad(Doctrine 2023)

1.6.8 Servidor Web:

Apache2: El Proyecto del Servidor HTTP Apache es un esfuerzo colaborativo de desarrollo de software destinado a crear una implementación de código fuente sólida, de calidad comercial, con muchas funciones y de libre acceso de un servidor HTTP (Web). El proyecto es gestionado conjuntamente por un grupo de

voluntarios ubicados en todo el mundo, que utilizan Internet y la Web para comunicar, planificar y desarrollar el servidor y su documentación relacionada. Este proyecto es parte de la Apache Software Foundation. Además, cientos de usuarios han aportado ideas, código y documentación al proyecto. Este archivo tiene como objetivo describir brevemente la historia del servidor HTTP Apache y reconocer a los numerosos contribuyentes.(Apache 2023)

XAMPP: El objetivo de XAMPP es crear una distribución fácil de instalar para desarrolladores que se están iniciando en el mundo de Apache. XAMPP viene configurado por defecto con todas las opciones activadas. XAMPP es gratuito tanto para usos comerciales como no comerciales.(Apachefriends 2023)Esta herramienta se empleó para configurar de manera más sencilla el servidor web apache.

WSO2 API Manager: es una solución de código totalmente abierto para la gestión de API de un extremo a otro en la nube, en entornos locales o híbridos. Viene con una licencia de software Apache versión 2.0 que lo hace de uso gratuito. Permite a los desarrolladores de API diseñar, publicar y administrar el ciclo de vida de las API y a los administradores de productos API para crear productos API a partir de una o más API. Alberga un portal para desarrolladores de aplicaciones que ayuda a crear y gestionar una comunidad de desarrolladores para sus API. Su puerta de enlace API nativa en la nube se utiliza para proteger, enrutar, controlar y monitorear su tráfico API de manera escalable.(API Manager 2023)Se empleo de medio de comunicación entre la aplicación y las aplicaciones externas de le empresa.

1.7 Conclusiones parciales

Después de haber realizado un análisis de los conceptos asociados al dominio del problema, una descripción de la entidad en la que se realizó la investigación, así como los antecedentes, la metodología y las herramientas utilizadas, se arribó a las siguientes conclusiones:

- No existe ningún sistema que resuelva las necesidades detectadas en el objeto de estudio.
- La combinación de herramientas, tecnologías y la metodología de desarrollo de software establecida por la XETID, es la apropiada para la realización del sistema ya que resuelve la situación problémica planteada en la investigación.
- El análisis del estado del arte permitió una mejor comprensión del objeto de estudio, estableciendo las bases para las siguientes fases de la investigación.

Capítulo 2: Diseño y construcción de la Solución Propuesta

Introducción

En este capítulo se detallan los elementos esenciales del sistema a realizar, a través de su modelo conceptual. Además, se describen las características que deberá cumplir el sistema según lo especificado en la metodología de desarrollo de software Prodesoft, correspondientes a la fase de modelación, se aclaran los requisitos funcionales y no funcionales que debe tener el mismo.

2.1 El modelo del proceso de negocio

El proceso se inicia cuando el usuario, previamente registrado en el proveedor de identidad de ENZONA accede a la aplicación (Bodega Digital) y solicita iniciar sesión para los cual debe especificar sus credenciales en el proveedor de identidad. Luego esta información es enviada al Sistema de Gestión de Roles el cual verifica los datos del usuario y envía a la aplicación información del perfil de usuario (Datos específicos dentro de la aplicación, rol y permisos). Esta comunicación entre el proveedor de identidad y la aplicación se realiza mediante servicios ApiRest gestionadas y publicadas en el WSO2.

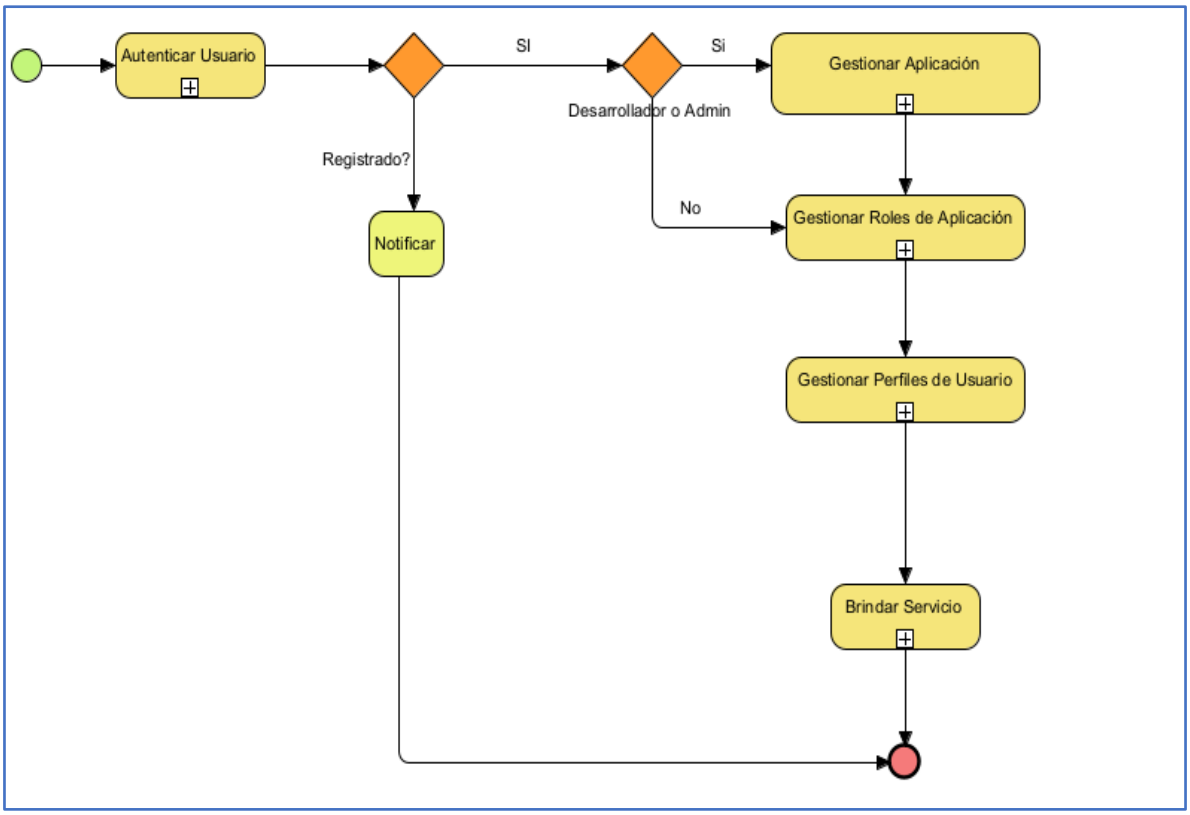


Figura 2 Modelo de Proceso del negocio

2.2 Modelo Conceptual

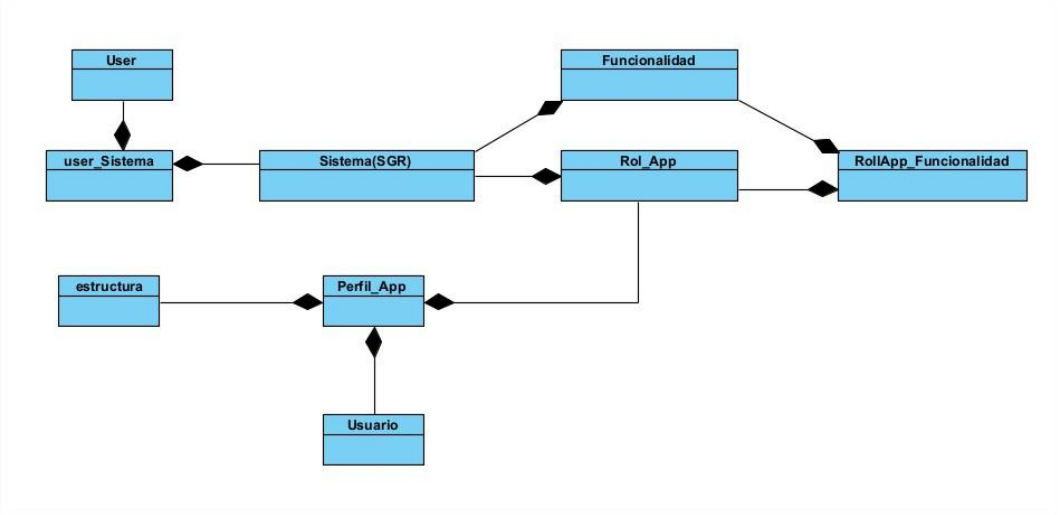


Figura 3 Modelo Conceptual

2.3 Requisitos

2.3.1 Requisitos funcionales

2.3.1 Configuración del sistema

2.3.1.1. Gestionar usuarios de sistema

R1. Adicionar usuario

R2. Editar usuario

R3. Mostrar datos de usuario

R4. Cambiar contraseña

R5. Habilitar usuario

R6. Deshabilitar usuario

2.3.1.2 Gestionar estructura

R7. Crear estructura

R8. Editar estructura

R9. Eliminar estructura

2.3.1.3 Gestionar composición de la estructura

R10. Adicionar entidad a la estructura

R11. Adicionar comercio a la estructura

R12. Eliminar comercio de la estructura

R13. Eliminar entidad de la estructura

2.3.1.4 Gestionar aplicaciones y subsistemas

R14. Adicionar aplicación

R15. Editar aplicación

R16. Eliminar aplicación

R17. Adicionar subsistema
R18. Editar subsistema
R19. Eliminar subsistema
R20. Adicionar funcionalidad
R21. Editar funcionalidad
R22. Eliminar funcionalidad

2.3.1.5 Gestionar usuarios por aplicaciones
R23. Asignar acceso a las aplicaciones
R24. Mostrar usuarios de la aplicación
R25. Ver datos de perfil de usuario
R26. Cambiar rol al usuario

2.3.1.6 Gestionar roles de aplicación
R27. Adicionar rol
R28. Editar rol
R29. Eliminar rol

2.3.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requisitos funcionales, es decir, una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser". (Proceso de Desarrollo y Gestión de Proyectos de Software, 2012, pág. 49) Para la presente investigación se seleccionaron los siguientes requisitos no funcionales:

Requisitos de usabilidad

El sistema debe ser fácil de utilizar por cualquier persona que tenga un conocimiento básico de trabajo en la web o de computación.

El sistema debe presentar una opción de ayuda sobre las principales funcionalidades que presenta.

El sistema debe implementarse lo más parecido posible a como se realiza el proceso en la actualidad para lograr una mejor comprensión y adaptación al mismo.

Requisitos de confiabilidad

Deben establecerse mecanismos que aseguren el reinicio del sistema ante diferentes fallos de forma rápida y eficiente.

Debe existir sistemas de respaldo eléctrico en los locales donde se encuentren los servidores.

Requisitos de seguridad y privacidad

El usuario debe autenticarse para acceder al sistema, dependiendo del nivel de acceso se presentarán las interfaces para cada usuario.

La información existente en el sistema será protegida contra actos ilícitos, de igual manera el origen y fuente de los datos.

2.4 Especificación de requisitos funcionales

A continuación, se presentan los diagramas de actividades de los principales requisitos funcionales.

Diagrama de actividades del requisito funcional Gestionar usuarios de sistema

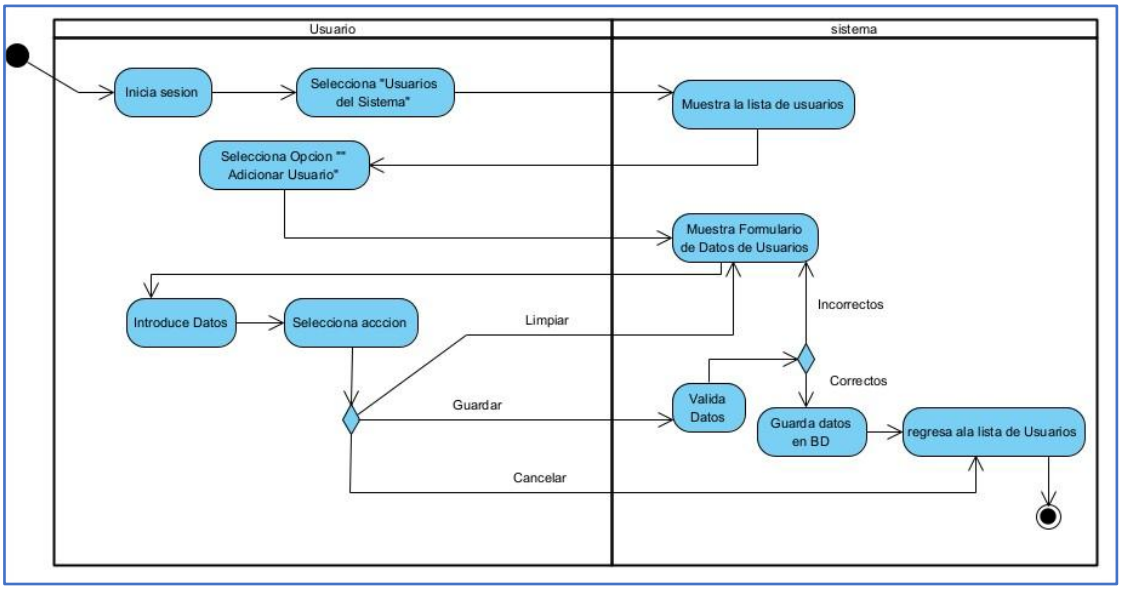


Figura 4 Diagrama de actividades Gestionar usuario del sistema

Diagrama de actividades del requisito funcional Gestionar estructura

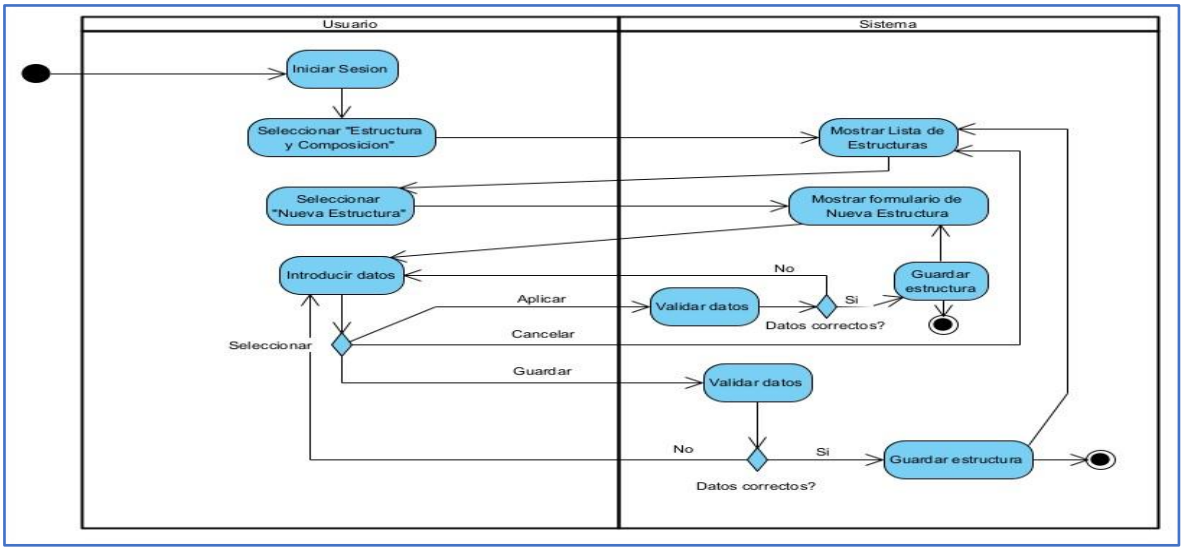


Figura 5 Diagrama de actividades Gestionar estructura

Diagrama de actividades del requisito funcional Gestionar usuarios por aplicaciones

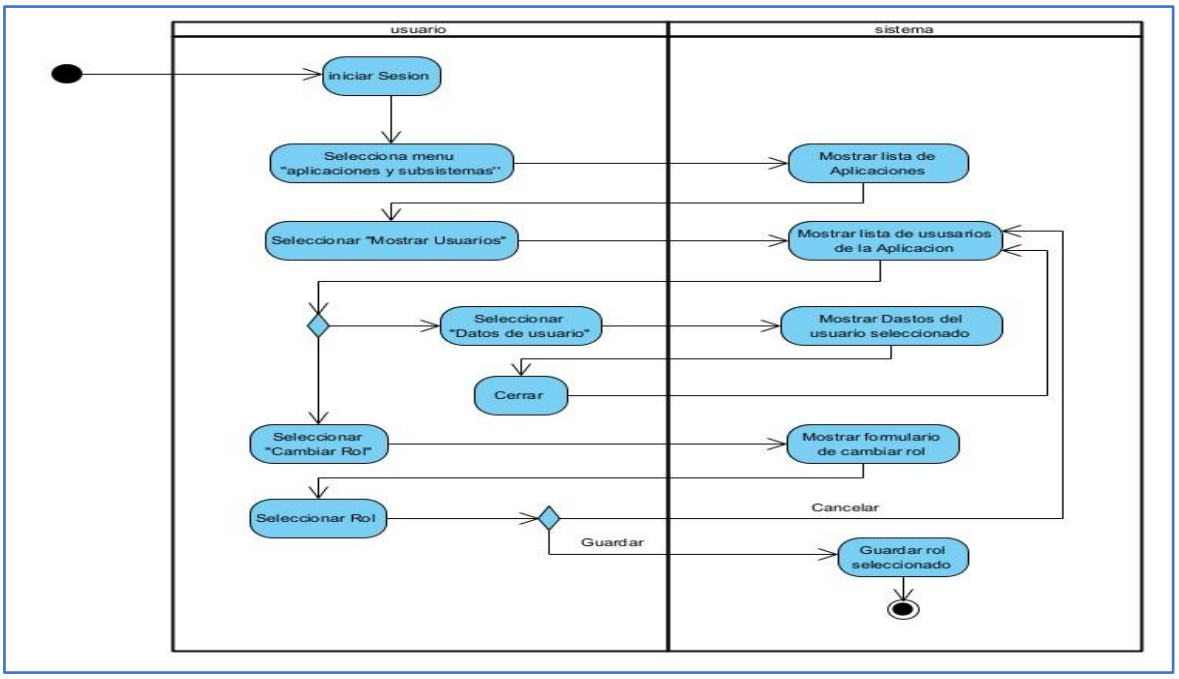


Figura 6 Diagrama de actividades Gestionar usuarios por aplicación

Diagrama de actividades del requisito funcional Gestionar aplicaciones y subsistemas

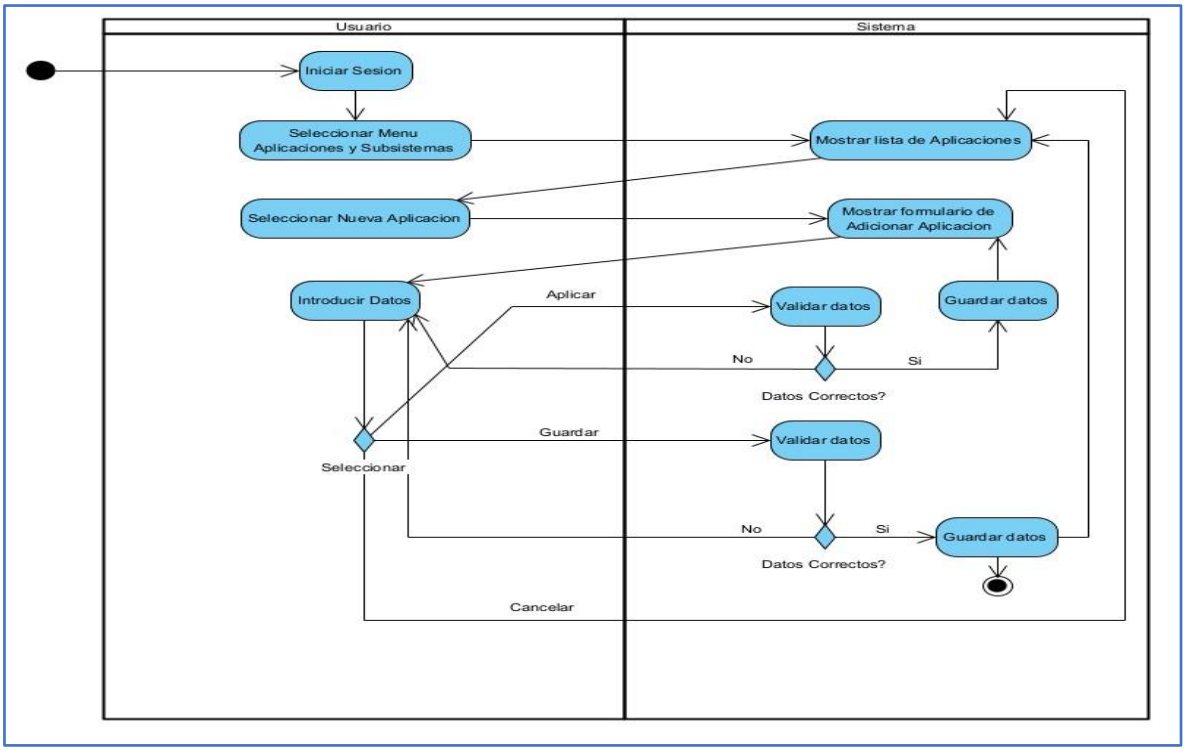


Figura 7 Diagrama de actividades Gestionar aplicaciones y subsistemas

Diagrama de actividades del requisito funcional Gestionar roles de aplicación

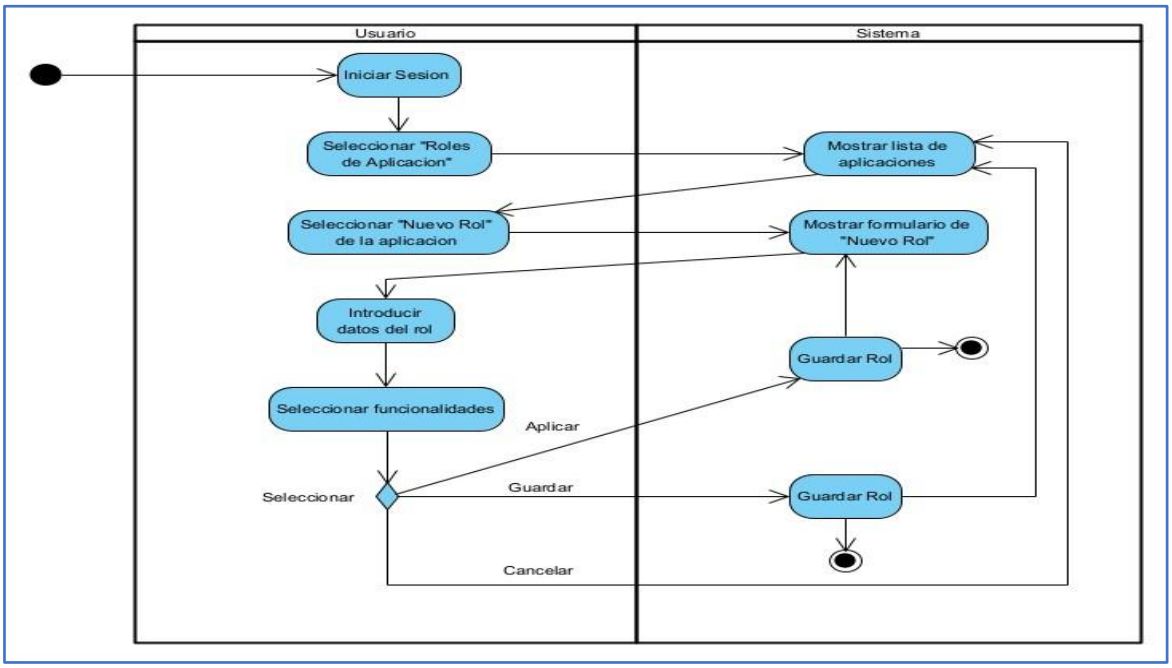


Figura 8 Diagrama de actividades Gestionar roles de aplicación

2.5 Diagrama de Casos de Uso

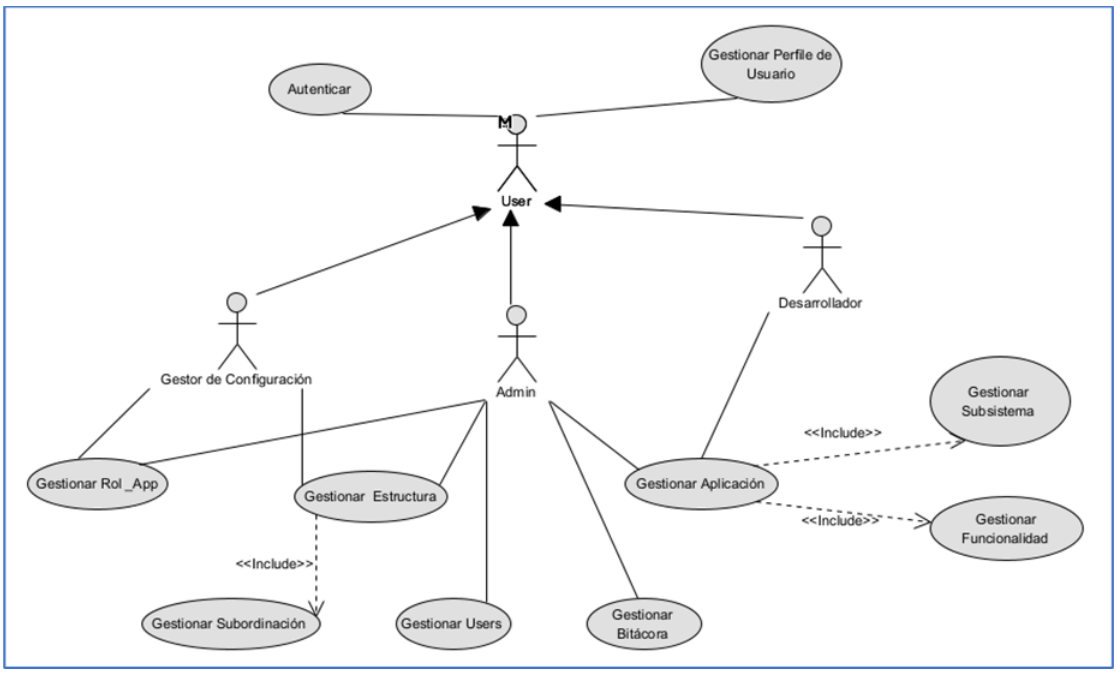


Figura 9 Diagrama de casos de uso

2.6 Implementación de la Arquitectura de Software

Según ProDeSoft (2012, pág. 53) "La Arquitectura de Software constituye un puente entre el requisito y el código, ocupando el lugar que en los modelos antiguos se reservaba para el diseño" o también se puede definir como "la organización fundamental de un sistema encarnado en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución."

2.6.1 Arquitectura

Modelo Vista Controlador (MVC). Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El estilo de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.(MVC 2023)

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el

propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.(MVC 2023)

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.(MVC 2023)

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.(MVC 2023)

Ventajas del MVC

Una separación total entre lógica de negocio y presentación. A esto se le pueden aplicar opciones como el multilenguaje, distintos diseños de presentación, etc. sin alterar la lógica de negocio. La separación de capas como presentación, lógica de negocio, acceso a datos es fundamental para el desarrollo de arquitecturas consistentes, reutilizables y más fácilmente mantenibles, lo que al final resulta en un ahorro de tiempo en desarrollo en posteriores proyectos.

Al existir la separación de vistas, controladores y modelos es más sencillo realizar labores de mejora como:

• Agregar nuevas vistas.

• Agregar nuevas formas de recolectar las órdenes del usuario (interpretar sus modelos mentales).

• Modificar los objetos de negocios bien sea para mejorar el performance o para migrar a otra tecnología.

• Las labores de mantenimiento también se simplifican y se reduce el tiempo necesario para ellas. Las correcciones solo se deben hacer en un solo lugar y no en varios como sucedería si tuviésemos una mezcla de presentación e implementación de la lógica del negocio.

• Las vistas también son susceptibles de modificación sin necesidad de provocar que todo el sistema se paralice. Adicionalmente el patrón MVC propende a la especialización de cada rol del equipo, por tanto, en cada liberación de una nueva versión se verán los resultados (MVC 2023).

2.7 Diseño de la Base de Datos

Teniendo como entrada el Modelo conceptual, la Especificación de la arquitectura de sistema y la Especificación de los requisitos de software se diseñan las tablas, sus atributos y relaciones agrupados por componentes ya sea por paquetes o colores delimitando cada uno de ellos, obteniendo como resultado el Modelo de datos". (ProDeSoft 2012).Este último es el que define cómo se modela la estructura lógica de una base de datos. Estos son entidades fundamentales para introducir la abstracción en una base de datos. Además, define cómo los datos se conectan entre sí y cómo se procesan y almacenan dentro del sistema(Glossaire

2023). Para el presente proyecto el modelo lógico obtenido se observa en la ilustración siguiente.

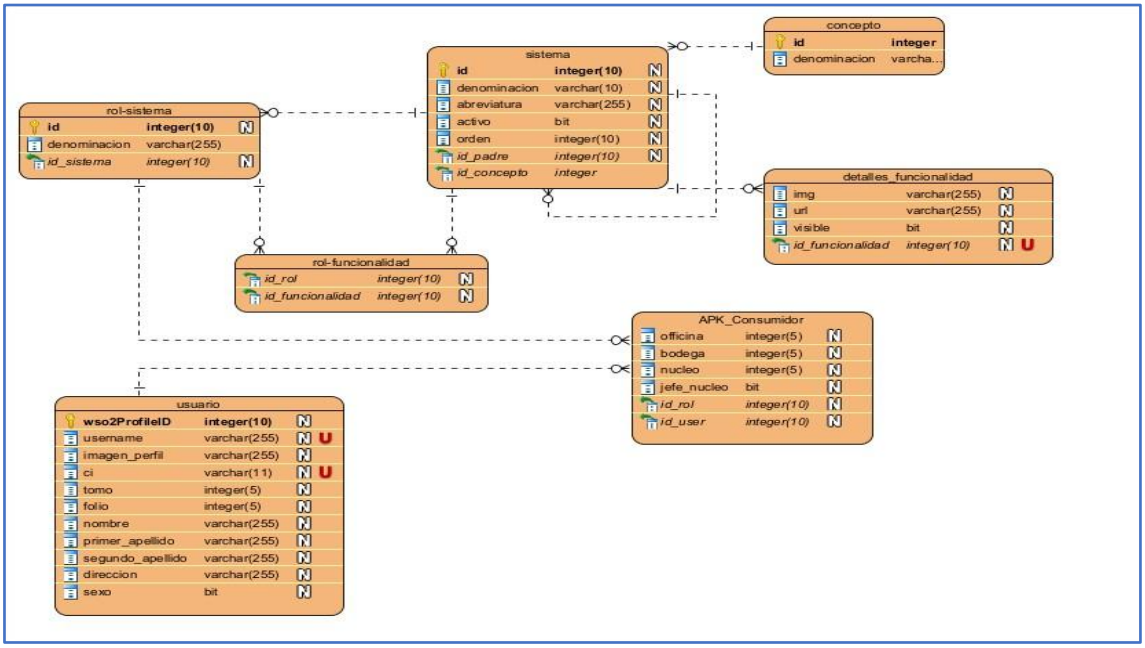


Figura 10 Diagrama de BD 1

El modelo físico de la base de datos se puede observar a continuación:

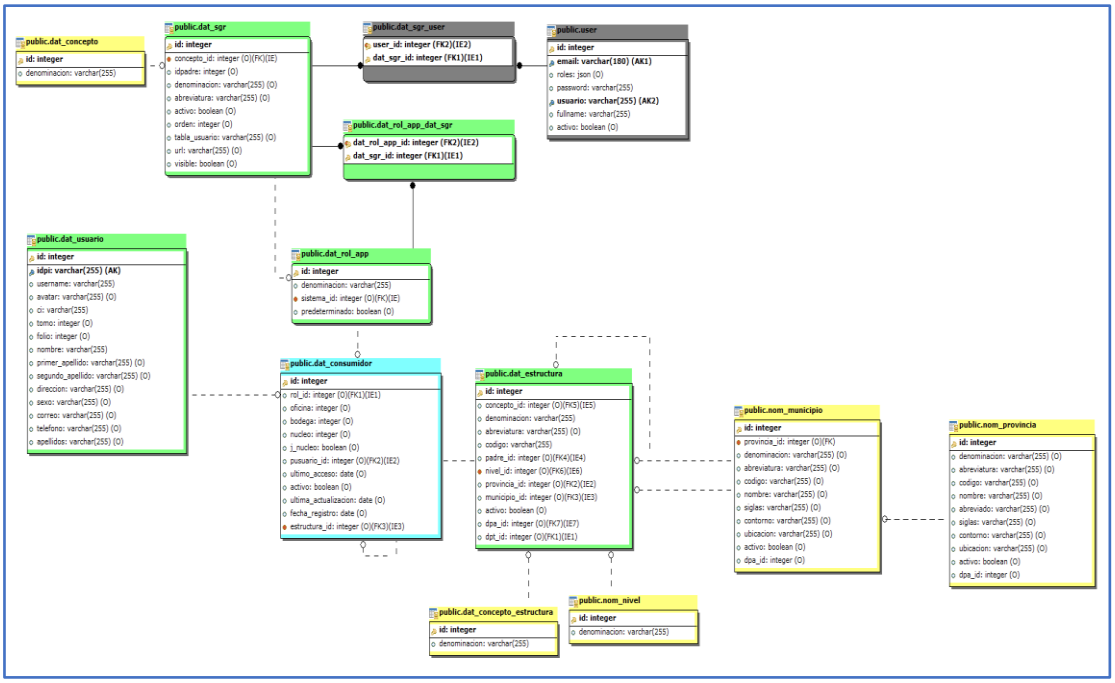


Figura 11 Diagrama físico de BD

2.8 Análisis de factibilidad

Para el análisis del costo y beneficios del sistema se utilizó la metodología de ProDeSoft donde desde el inicio se estima de forma empírica la duración de la implementación de cada uno de los requisitos, basado en la experiencia del programador en el trabajo con el lenguaje de programación, el entorno de desarrollo, el conocimiento sobre el tema de investigación y las técnicas de programación necesarias para resolver el problema. Para esto es necesario conocer el tiempo de desarrollo de cada requisito y la cantidad de trabajadores que participan para estimar si resulta beneficioso su desarrollo.

2.8.1 Costo

Para la estimación del costo del software se empleó la fórmula propuesta por la metodología, para lo cual primeramente se tiene en cuenta los requisitos funcionales a desarrollar, así como su prioridad, complejidad y tiempo estimado de desarrollo, lo que se detalla en la siguiente tabla, en la cual se tiene en cuenta que el tiempo total estimado para la realización del proyecto es de 6 meses, que serían 24 semanas:

No.	**Nombre del requisito**	**Prioridad**	**Complejidad**	**Tiempo de desarrollo (semanas)**
1	Gestionar usuarios del sistema	Alta	Media	3
2	Gestionar estructura	media	media	2
3	Gestionar composición de la estructura	media	media	2
4	Gestionar aplicaciones	alta	alta	3
5	Gestionar usuarios por aplicaciones	alta	media	2
6	Gestionar roles de aplicaciones	alta	media	3
7	Gestionar Metadatos de Aplicación	alta	alta	5
8	Gestionar funcionalidad	alta	media	2
9	Gestionar Subsistemas	alta	media	2

Tabla 1 Costo

Luego se determina que:

Costo diario de un trabajador= Tarifa horaria (MT) * Trabajador * Tiempo diario.

= $20.00 * 1 trabajador * 8 horas.

= $160.00.

Tomando en cuenta que un trabajador de la entidad trabaja 20 días al mes

= $160.00. *20

= $3 200.00.

Costo mensual de un trabajador = $3 200.00.

Teniendo en cuenta un tiempo estimado de aproximadamente 6 meses para el desarrollo del proyecto se determina:

Costo total= $3 200.00* 6 meses.

= $ 19,200.00

El costo de desarrollo del sistema fue $19 200.00 aproximadamente, lo que en materia económica constituye una cifra moderada de dinero con respecto a los beneficios que se evidencian a continuación.

2.8.2 Análisis de costo y beneficios

Anteriormente se calculó el costo del sistema y se analizaron los beneficios que representa el mismo, lo que permite concluir que el sistema es positivo para gestionar de forma integrada y segura la información de módulos y funcionalidades, y los roles del negocio de cada una de las aplicaciones, así como la gestión de acceso de los usuarios mediante la asignación de roles y entidades.

2.9 Patrones

Los patrones de diseño según (Martínez 2023)son una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas. Los patrones no se proponen descubrir ni expresar nuevos principios de la ingeniería del software. Todo lo contrario: intentan codificar el conocimiento, las expresiones y los principios ya existentes: cuanto más trillados y generalizados, mucho mejor.

Para la implementación de la presenta investigación se utilizaron como patrones, los siguientes:

- **Singleton**: Este patrón se encuentra dentro de los patrones de tipo creación, que son aquellos que abstraen el proceso de creación de instancias de los clientes que lo utilizan. Gracias al patrón Singleton, se puede hacer un sistema independiente de la forma en la que se crean las instancias, además, se hace independiente de la forma en la que se componen y representan las mismas. Esto es debido a que oculta, a los clientes, el proceso de creación y asociación de las instancias. El patrón Singleton es aplicable en cuando, únicamente debe de existir una instancia de una clase y debe ser accesible desde un punto conocido. La instancia única debe ser extensible por los clientes mediante herencia sin necesidad de modificar su código(Refactoring.guru 2023)

- **Constructor**: Como Patrón de diseño, el Constructor, es clasificado como creacional y es usado para permitir la creación de una variedad de objetos complejos desde un objeto fuente (Producto). El objeto fuente se compone de una variedad de partes que contribuyen individualmente a la creación de cada objeto complejo a través de un conjunto de llamadas a interfaces comunes de la clase Constructor Abstracto. Tiene como ventajas que el algoritmo para la creación de objetos complejos sea independiente de las partes que construyen el objeto y cómo son ensambladas y que el proceso de construcción pueda tener diferentes representaciones para el objeto que está construido(Builder Pattern 2023)

2.10 Diagrama de clases

Un Diagrama de Clases de Diseño muestra la especificación para las clases software de una aplicación. Incluye las clases, asociaciones y atributos; las interfaces, con sus operaciones y constantes; los métodos y las dependencias.

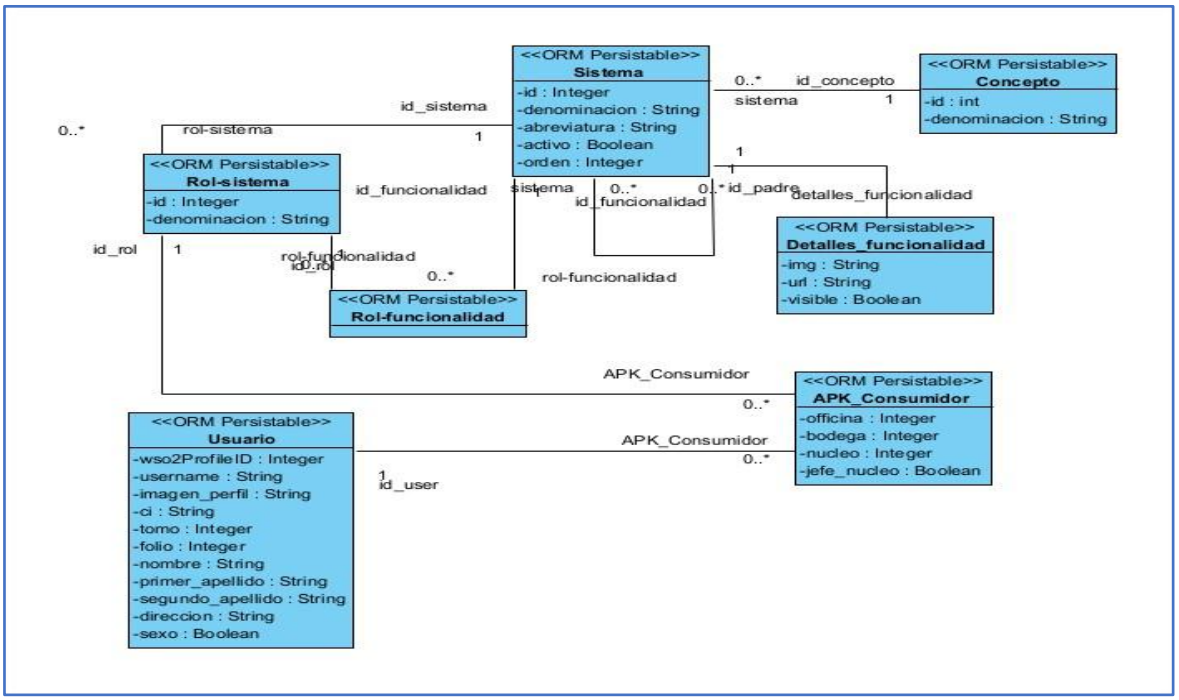


Figura 12 Diagrama de clases

2.11 Diagramas de secuencia

Los diagramas de secuencia son aquellos diagramas que permiten mostrar la interacción entre los usuarios, la interfaz y las instancias de los objetos en el sistema. Ellos proveen un mapa secuencial del paso de los mensajes entre los objetos a lo largo del tiempo. (ProDeSoft 2012)

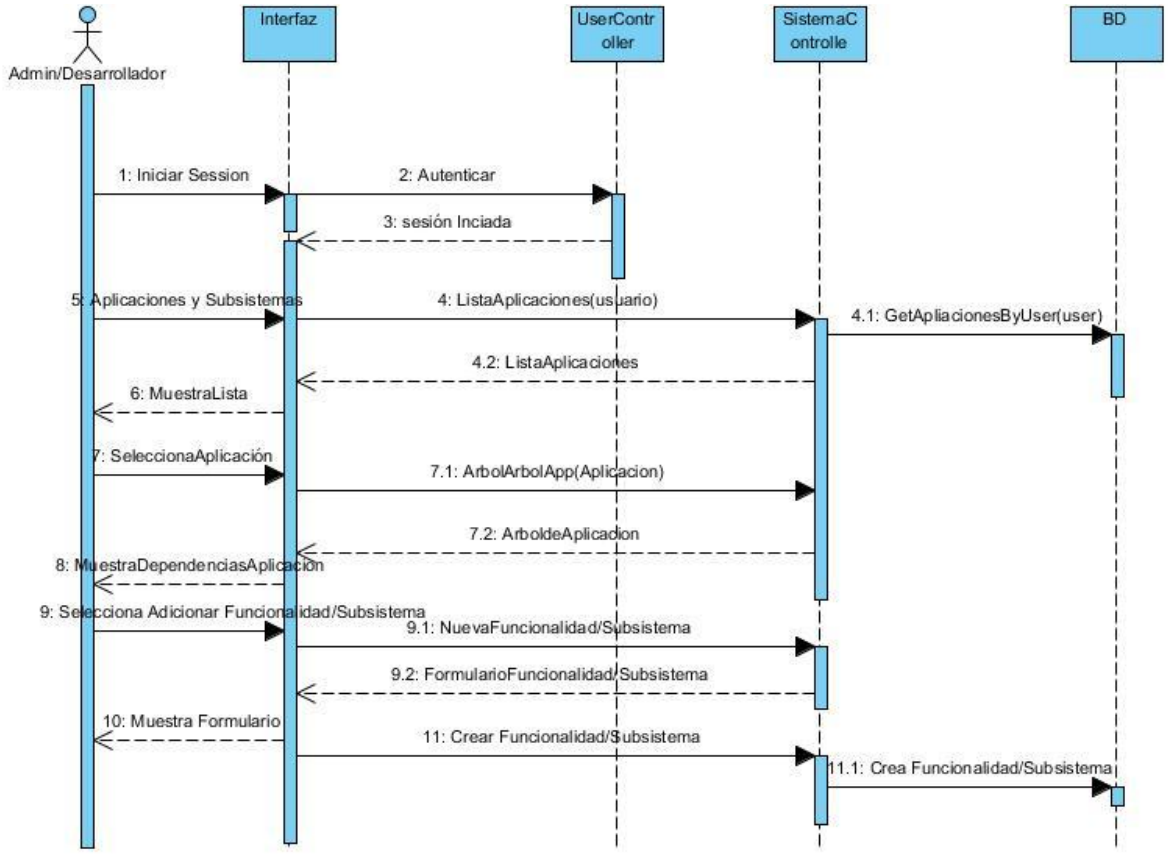


Figura 13 Diagrama de Secuencia

2.12 Seguridad

El sistema deberá implementar diversas políticas de seguridad para garantizar la confidencialidad, integridad y disponibilidad de la información que maneje. Para ello se utilizará una seguridad RBAC o Control de acceso basado en roles, el cual permite administrar quién tiene acceso a cada recurso y que pueden hacer con esos recursos, es decir, separa las tareas y otorga solo la cantidad de accesos que los usuarios necesitan para realizar sus trabajos (Microsoft 2023a). Además, tendrá en cuenta el desarrollo de una interfaz fácil de usar, sencilla y amigable; así como una ayuda general, que guíe al usuario durante su trabajo en el sistema. Por otra parte, el framework utilizado para el acceso a los datos, en este caso Doctrine, al igual que la mayoría de los ORM, presenta su propia capa de seguridad para la protección contra los ataques más comunes, como son las inyecciones SQL.

2.13 Conclusiones parciales

Después de haber realizado todos los diagramas especificados por Prodesoft y realizado el software bajo esas especificaciones, se arribó a las siguientes conclusiones:

• La modelación de todos los procesos que intervinieron durante el desarrollo del sistema proporcionó una visión más completa del producto deseado.
• La realización del modelado del proceso permitió un mayor entendimiento del negocio, facilitando la posterior programación del mismo.
• Los diagramas de clases permitieron conocer los requisitos con un nivel de profundidad más amplio, aclarando todos los detalles necesarios para una buena implementación de los mismos.
• Los diagramas de secuencia permitieron el proceso de implementación del sistema de una forma más clara.
Se confirmó que se requieren varias iteraciones de pruebas de caja negra para garantizar que el producto final cumpla con las funcionalidades exigidas por el cliente, aunque en este caso solo fueron necesario 2 iteraciones

Capítulo 3

En este capítulo se presentan los resultados que se obtienen al aplicar la metodología explicada en el capítulo 2, así como las pruebas al software realizadas, que permiten conocer el grado de calidad del producto, y de esta forma, comprobar si el sistema es capaz de realizar todas las funcionalidades detallas anteriormente. Se prueban las características más importantes de la aplicación con el fin de verificar la fiabilidad y calidad de la aplicación como un todo.

3.1 Descripción de la solución

La solución obtenida es una aplicación la cual cuando el usuario, previamente registrado en el proveedor de identidad de ENZONA accede a la aplicación (Bodega Digital) y solicita iniciar sesión para los cual debe especificar sus credenciales en el proveedor de identidad. Luego esta información es enviada al Sistema de Gestión de Roles el cual verifica los datos del usuario y envía a la aplicación información del perfil de usuario (Datos específicos dentro de la aplicación, rol y permisos). Esta comunicación entre el proveedor de identidad y la aplicación se realiza mediante servicios ApiRest gestionadas y publicadas en el WSO2.

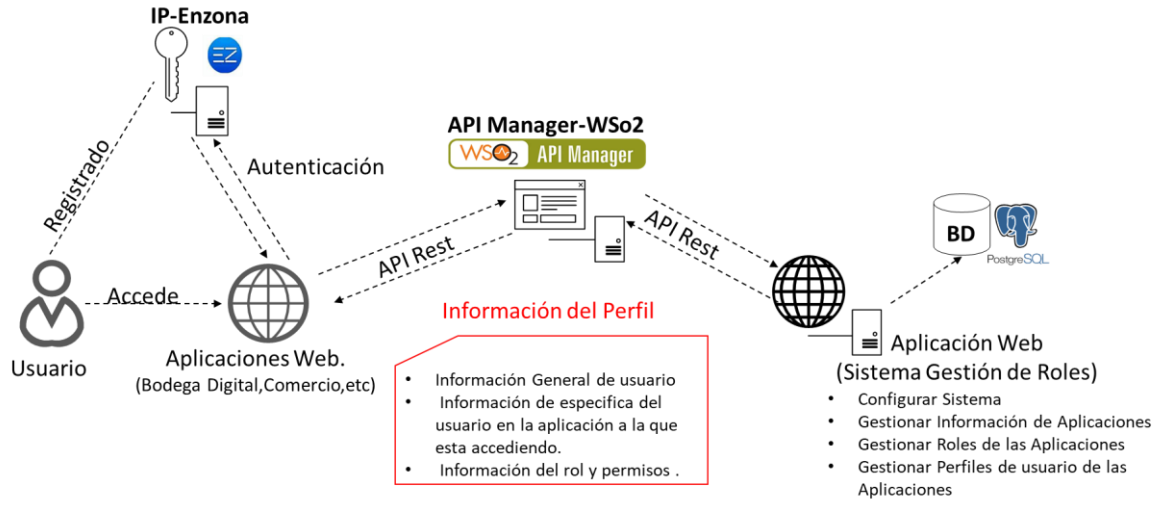
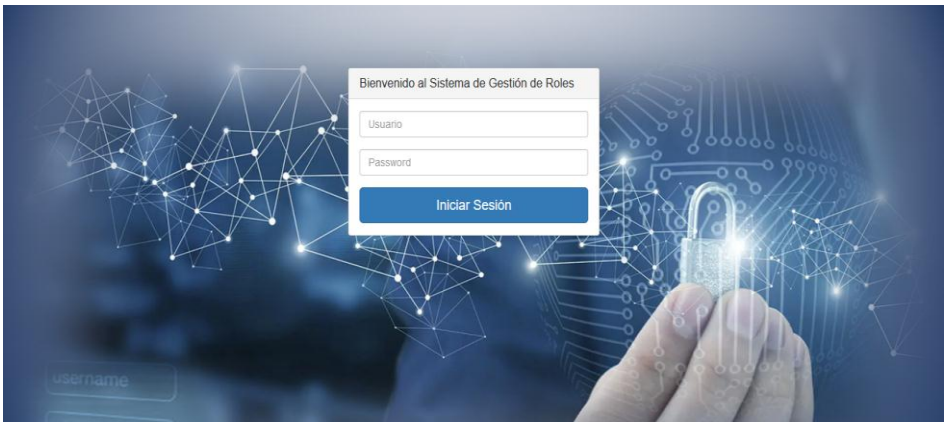


Figura 14 Diagrama de Componentes

3.2 Resultados obtenidos

3.2.1 Interfaces de usuario

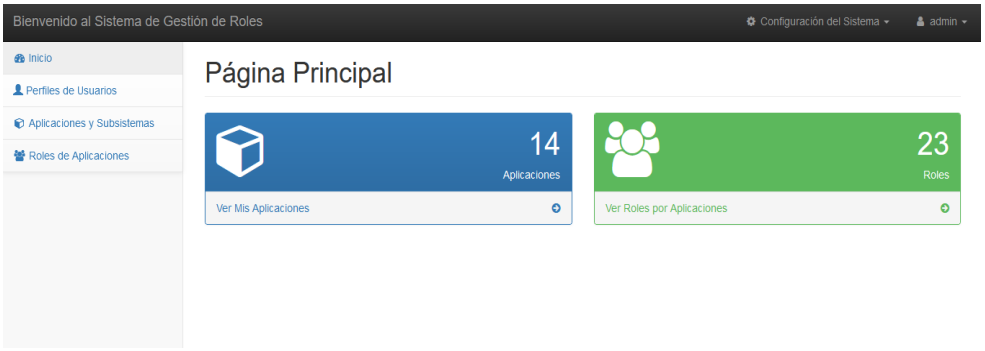
En este apartado se muestran algunas de las interfaces del sistema. En todas se puede apreciar un diseño sencillo, usable para el cliente.



Vista 1 Inicio de sesión

Se muestra la interfaz de inicio de sesión donde se piden las credenciales del usuario para acceder al sistema. Se accede mediante un usuario y contraseña previamente creados por un usuario con permisos de administración en el sistema.

1. Dashboard



Vista 2 Dashboard

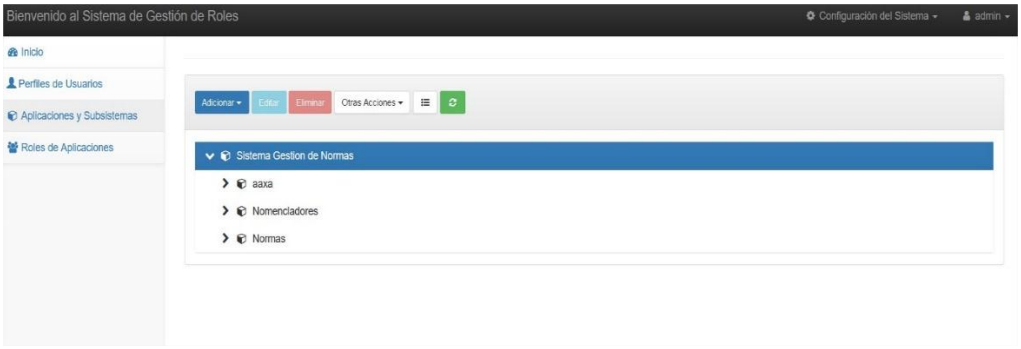
Se muestra la página inicial del sistema donde se puede acceder a los diferentes menús disponibles para el usuario.

2. Aplicaciones y Subsistema

2.1. Lista

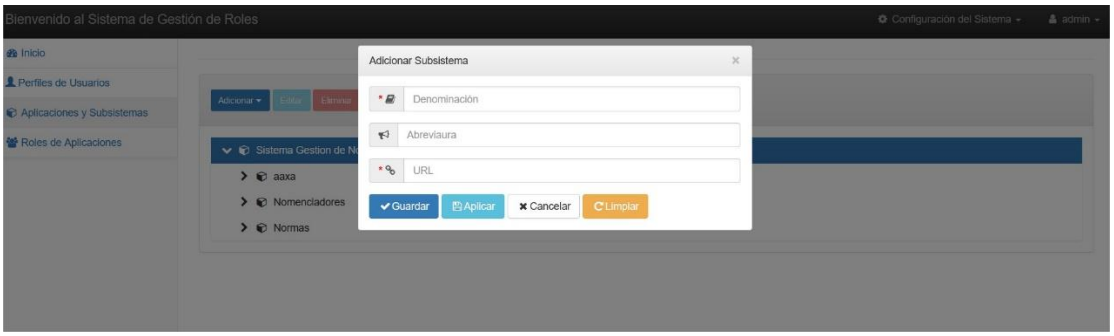
Vista 3 Lista de aplicaciones

2.2. Gestionar Información: Gestionar Información: Al hacer click en el botón de gestionar información pasamos a la interfaz donde se nos muestra un listado de subsistemas y funcionalidades que posee la aplicación. Desde este menú se puede añadir, editar y eliminar subsistemas y funcionalidades.



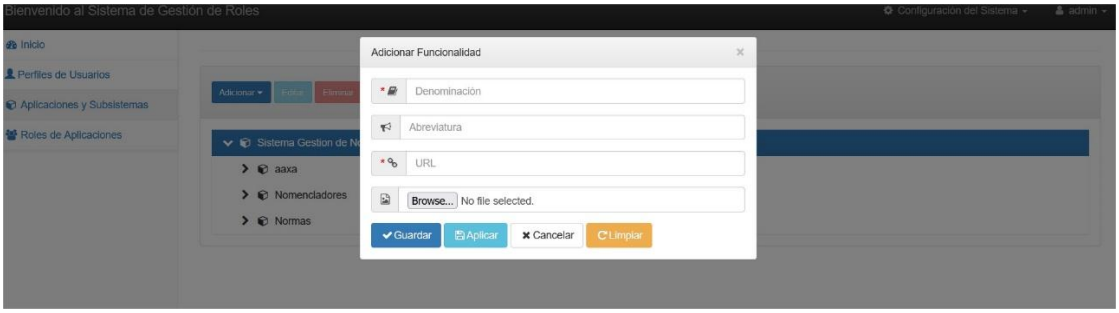
Vista 4 Gestionar Información de aplicación

2.2.1. Add Subsistema: ejemplo de vista de añadir un subsistema.



Vista 5 Adicionar subsistema

2.2.2. Add Funcionalidad: ejemplo de vista de añadir una funcionalidad.



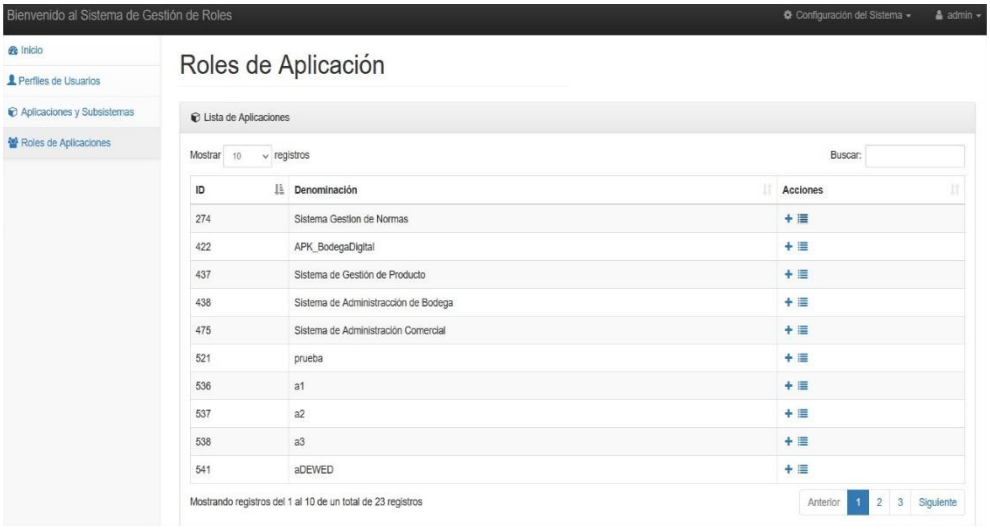
Vista 6 Adicionar Funcionalidad

2.3. Editar: ejemplo de vista de editar.

Vista 7 Editar

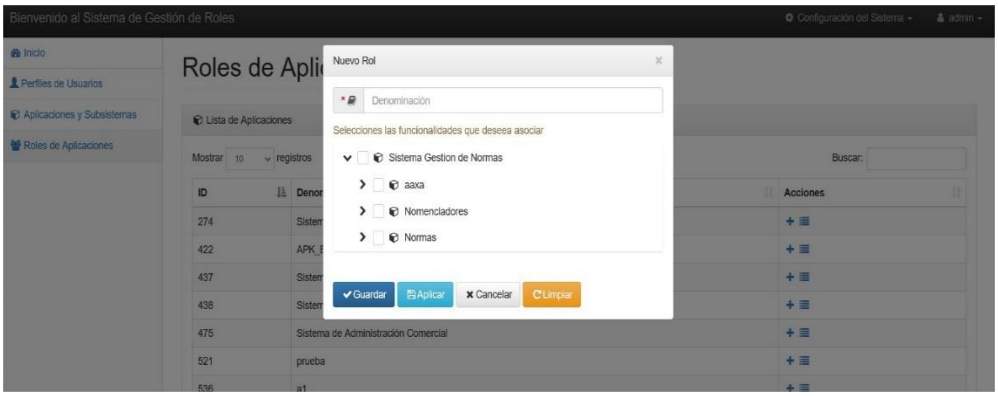
3. Roles y Aplicaciones

3.1. Listar App: Desde esta vista podemos añadir un nuevo rol a la aplicación deseada o mostrar un listado de los roles existentes por aplicación.



Vista 8 Listar app

3.2. Nuevo Rol: ejemplo de vista de crear nuevo rol asignándole un subsistema y funcionalidades registradas previamente.

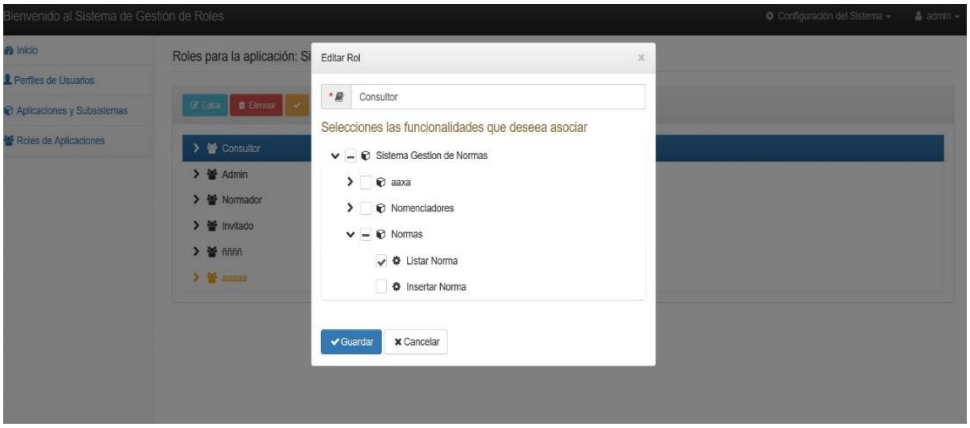


Vista 9 Adicionar Nuevo Rol

3.3. Lista Roles: Ejemplo de vista de un listado de roles registrados en una aplicación.

Vista 10 Listar Rol de app

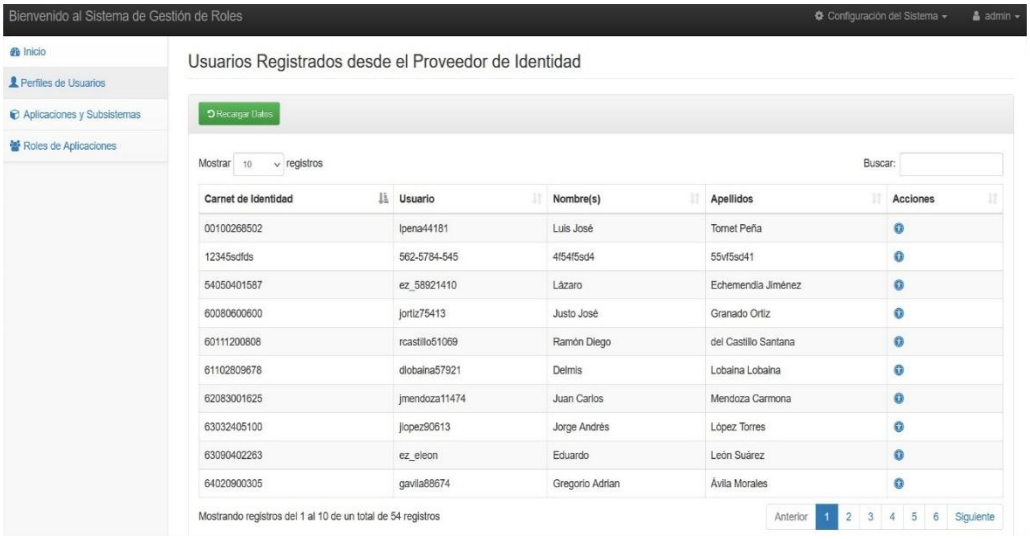
3.3.1. Editar Rol



Vista 11 Editar rol

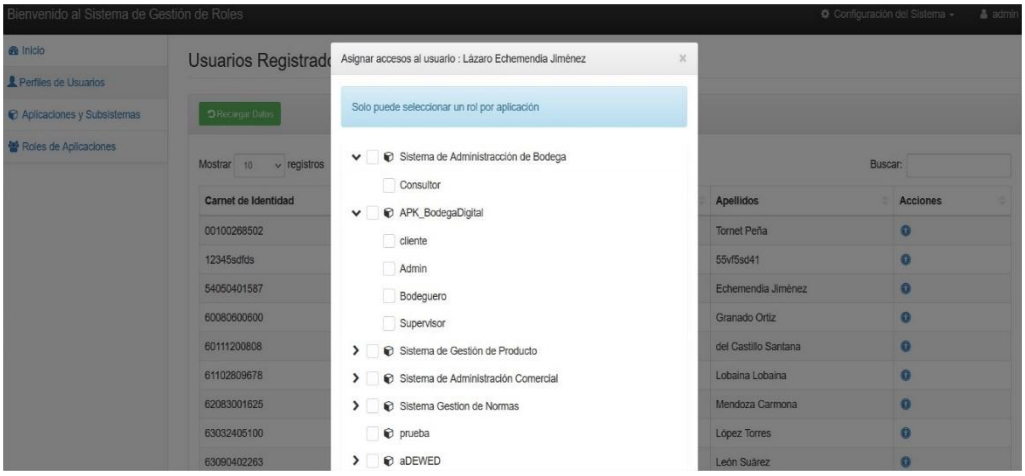
4. Perfiles de Usuario

4.1. Listar: listado de usuarios registrados a las aplicaciones mediante el proveedor de identidad Enzona.



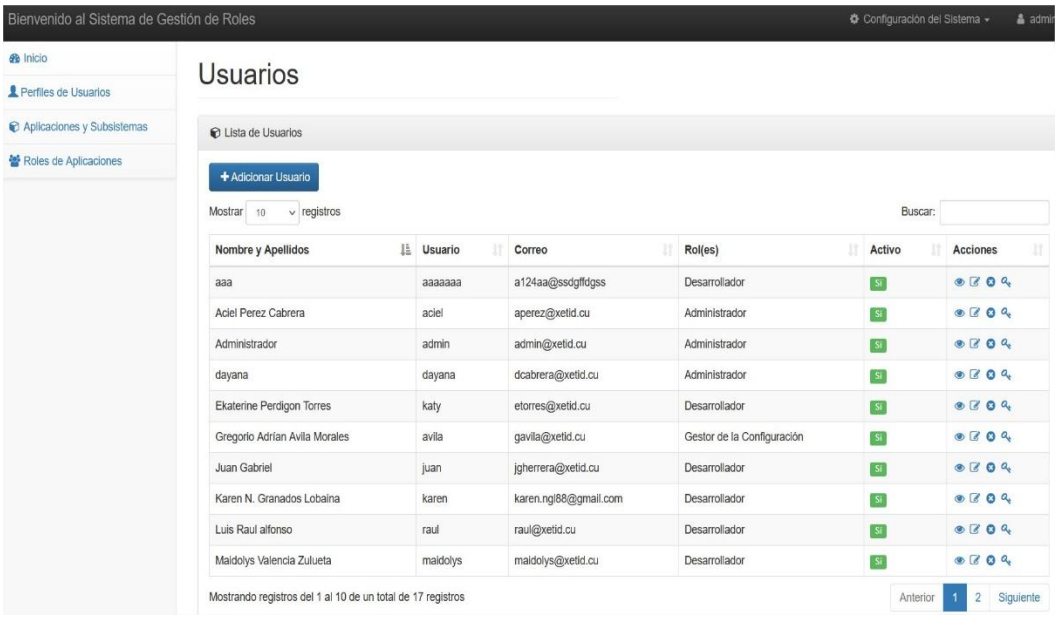
Vista 12 Lista de usuarios registrados por enzona

4.2. Asignar Acceso



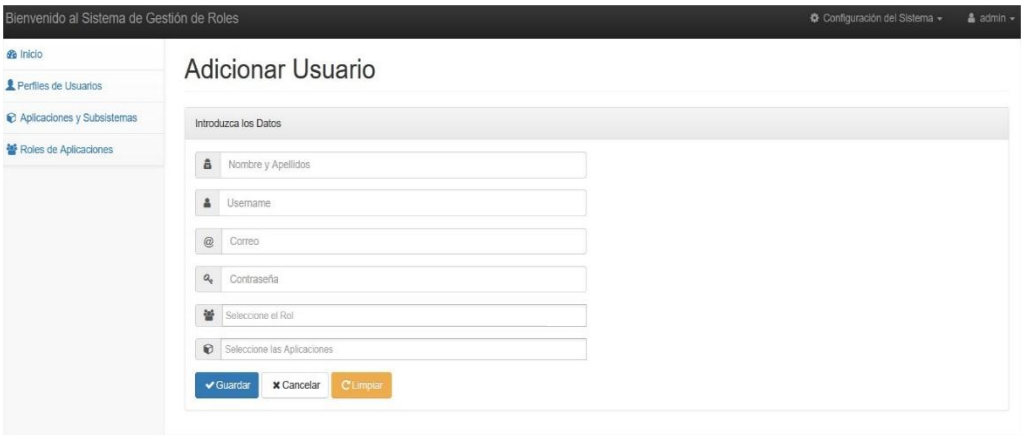
Vista 13 Asignar acceso al usuario

5. Configuración del Sistema/Usuarios del sistema: En esta vista se muestran los usuarios registrados en esta aplicación para poder acceder a la misma debe estar logueado como administrador. Desde este menú se puede adicionar nuevos usuarios, ver la información de los usuarios ya registrados y editar los mismos, así como habilitar o deshabilitar a los usuarios, también podemos cambiar la clave de acceso de los usuarios al sistema.



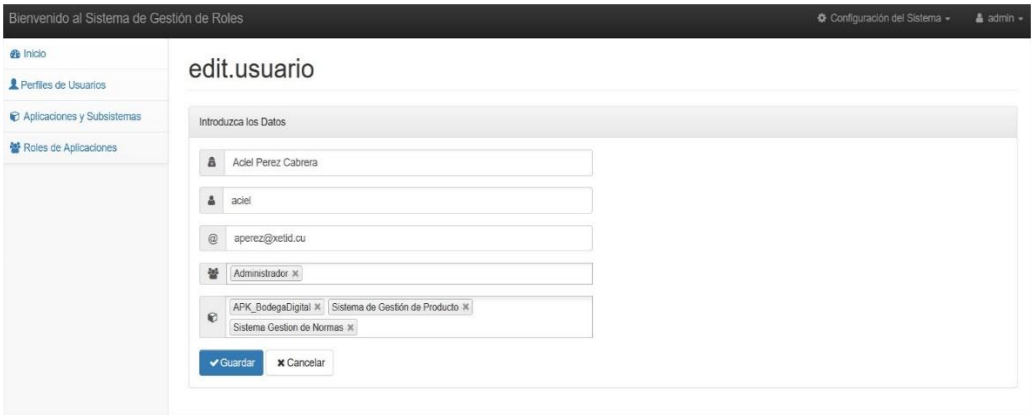
Vista 14 Lista de Usuarios del sistema

5.1. Nuevo Usuario: Ejemplo de vista de adicionar usuario.



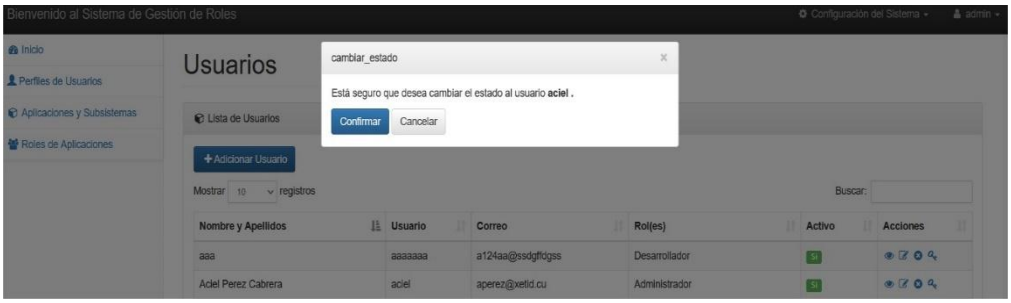
Vista 15 Adicionar usuario nuevo

5.2. Editar: Ejemplo de editar datos de usuario.



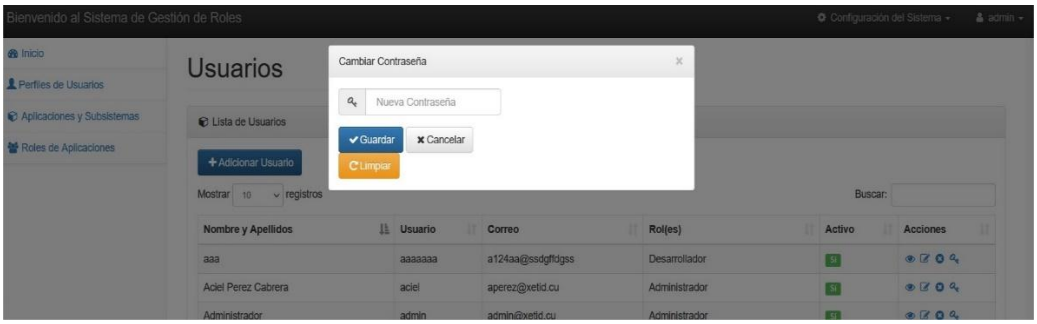
Vista 16 Editar datos de usuario

5.3. Cambiar Estado: Ejemplo de alerta al solicitar cambiar el estado de un usuario.



Vista 17 Cambiar estado del usuario

5.4. Cambiar Contraseña: Ejemplo de cambio de contraseña.

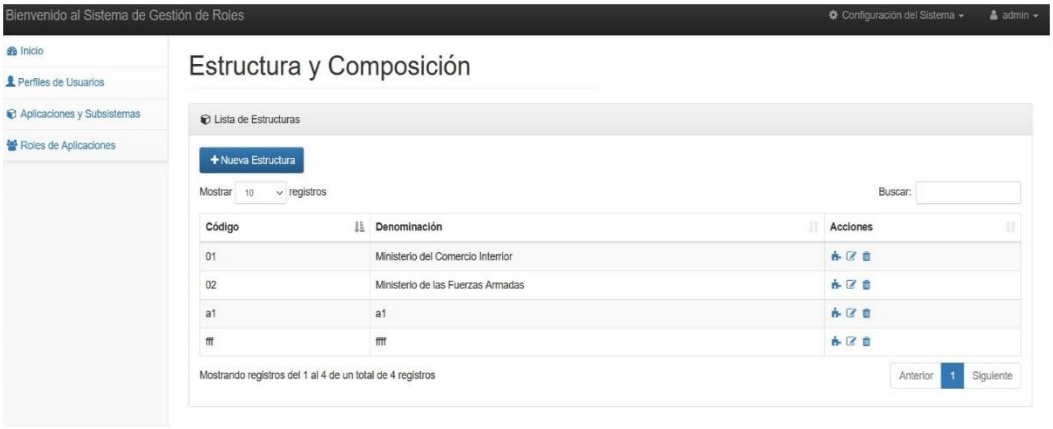


Vista 18 Cambiar contraseña de usuario

6. Configuración del Sistema/Estructura y Composición:

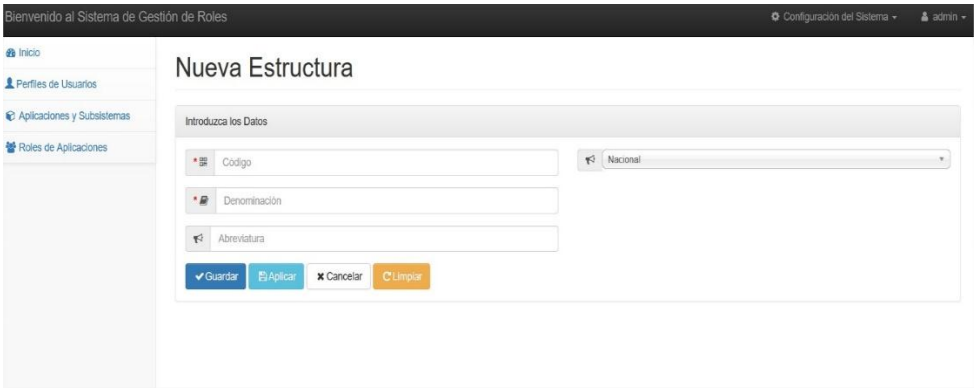
En este menú se pueden observar las estructuras registradas en el sistema, añadir nuevas estructuras, editar los datos de las estructuras previamente registradas, también se puede gestionar la composición de cada una de ellas.

6.1. Listar



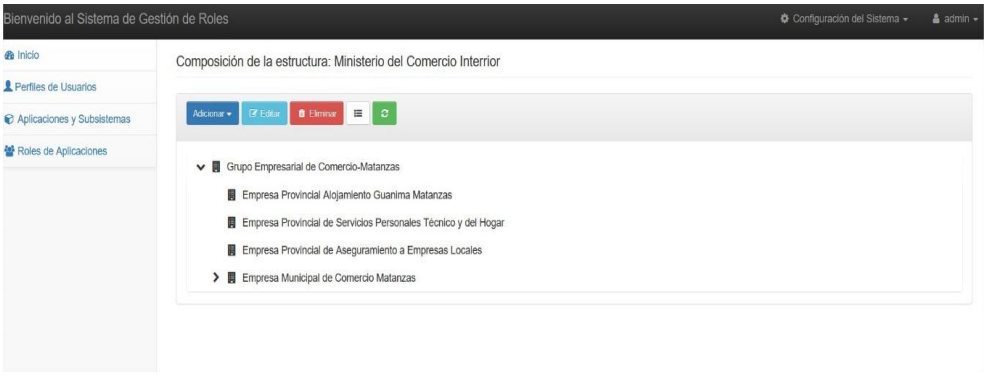
Vista 19 Lista de estructuras

6.2. Nueva Estructura: Ejemplo de añadir nuevas estructuras.



Vista 20 Adicionar nueva estructura

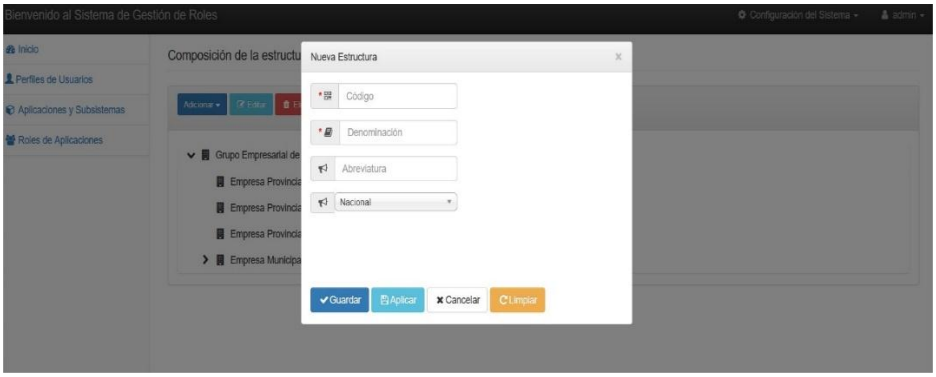
6.3. Gestionar Composición: Ejemplo de gestionar la composición de una estructura desde donde se nos permite añadir estructuras pertenecientes a la seleccionada, añadir comercio, editar la estructura o eliminar.



Vista 21 Gestionar composición

6.3.1. Add

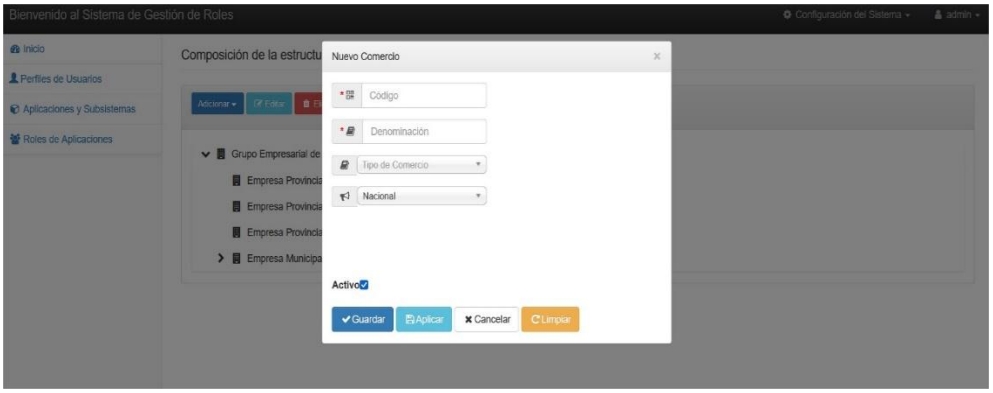
Entidad: Ejemplo de añadir estructura.



Vista 22 Agregar entidad

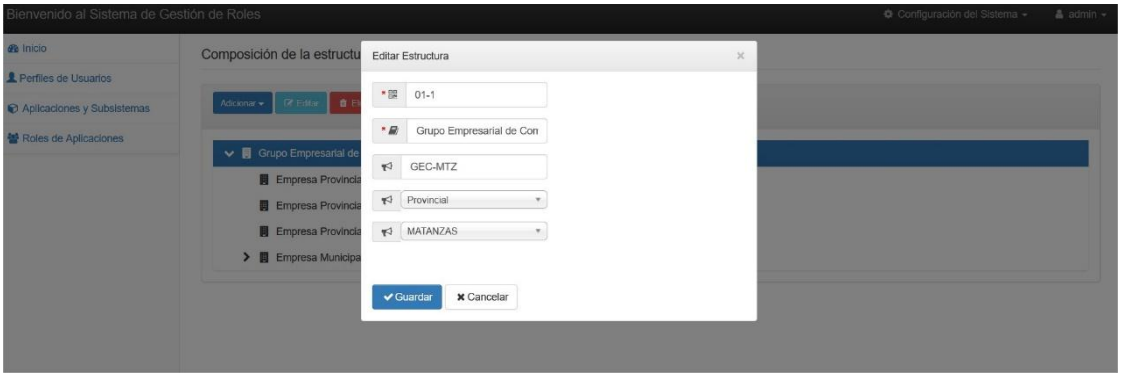
6.3.2. Add

Comercio: Ejemplo de Añadir comercio.



Vista 23 Agregar comercio

6.4. Editar: Ejemplo de Editar una estructura.



Vista 24 Editar

3.3 Pruebas realizadas y resultados

Las pruebas del software son la actividad más común de control de la calidad realizada en los proyectos para asegurar el correcto funcionamiento del software. Tienen como objetivos la verificación de la correcta implementación de los requisitos explícitamente establecidos, la adecuada integración de los componentes que conforman el sistema y la ejecución de casos de prueba que permitan detectar el mayor número de No conformidades y corregirlas antes de la entrega del software al cliente. Estas reducen la probabilidad de que aparezcan defectos ocultos en el software, pero incluso si no se encuentra ningún defecto, nunca será una garantía de su corrección. (ProDeSoft 2012) Sus objetivos principales son:

- Probar si el software no hace lo que debería hacer.
- Probar si el software hace lo que no debería hacer.

3.3.1 Pruebas de aceptación

Las pruebas de aceptación son las últimas pruebas realizadas donde el cliente prueba el software y verifica que cumpla con sus expectativas. Estas pruebas generalmente son funcionales y se basan en los requisitos definidos por el cliente y deben hacerse antes de la salida a producción. Esta termina de definir el nivel de calidad de la aplicación y le permite conocer al equipo qué tan bien supo interpretar los requerimientos del usuario. (Wordpress.com, 2017)

Estas pruebas se realizaron para cada uno de los requisitos funcionales del software y a continuación se presentan algunos de las realizadas para el requisito gestionar estructura, agregar nueva estructura, editar estructura, eliminar estructura, el resto se realizó de una forma similar. Estas son elaboradas por el propio equipo de desarrollo y sus resultados son guardados en tablas que cuentan con los campos que se exponen a continuación:

Prueba de aceptación	
Código caso de prueba: RF-2-P1	RF: Gestionar Estructura

Nombre del caso de prueba: Test agregar nueva estructura
Responsable: Jahaziel Betancourt Rodríguez
Descripción: Adicionar nueva estructura desde las configuraciones del sistema.
Condiciones de ejecución: El usuario logueado para esta operación debe ser Admin o Gestor de Configuración
Pasos de Ejecución: Seleccionar configuración del sistema. Seleccionar estructura y composición. Seleccionar nueva estructura.
Resultados Esperados: Que se muestre la opción nueva estructura y el formulario perteneciente a esta operación.
Evaluación: Satisfactoria

Tabla 2 Prueba1

Prueba de aceptación	
Código caso de prueba: RF-2-P2	RF: Gestionar Estructura
Nombre del caso de prueba: Test editar estructura	
Responsable: Jahaziel Betancourt Rodríguez	
Descripción: Editar una estructura desde las configuraciones del sistema.	
Condiciones de ejecución: El usuario logueado para esta operación debe ser Admin o Gestor de Configuración	
Pasos de Ejecución: Seleccionar configuración del sistema. Seleccionar estructura y composición. Seleccionar Editar estructura.	
Resultados Esperados: Que al editar los datos de una estructura se modifiquen los valores correctamente.	
Evaluación: Satisfactoria	

Tabla 3 Prueba 2

Prueba de aceptación	
Código caso de prueba: RF-2-P3	RF: Gestionar Estructura
Nombre del caso de prueba: Test eliminar estructura	
Responsable: Jahaziel Betancourt Rodríguez	
Descripción: Eliminar una estructura desde las configuraciones del sistema.	
Condiciones de ejecución: El usuario logueado para esta operación debe ser Admin o Gestor de Configuración	
Pasos de Ejecución: Seleccionar configuración del sistema. Seleccionar estructura y composición. Seleccionar eliminar estructura.	
Resultados Esperados: Que al eliminar la estructura se muestre el mensaje de confirmación correctamente y en caso de tener un usuario asignado a la estructura no se pueda eliminar y se muestre el mensaje.	
Evaluación: Satisfactoria	

Tabla 4 Prueba 3

3.3.2 Pruebas de caja negra

Las pruebas de caja negra son una técnica en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna del código, detalles de implementación o escenarios de ejecución internos en el software. En ellas, el enfoque está dado solamente en las entradas y salidas del sistema, sin preocuparse en la estructura interna del programa. Para ellos se basa en los requerimientos de software y especificaciones funcionales. (Morrillo, 2017) Estas se enfocan en buscar:

Funciones incorrectas o ausentes.

- Errores de interfaz.
- Errores en estructuras de datos.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Se realizó un caso de prueba para cada requisito implementado. A continuación, se muestra el caso de prueba realizado al requisito Crear estructura, que describe cada uno de los escenarios que pueden existir ante las posibles acciones realizas por el usuario. De manera similar se realizaron los casos de prueba a los requisitos restantes.

Nombre del requisito	Descripción general	Escenarios de Pruebas	Flujo del Escenario
Crear estructura	Se Crea una Nueva estructura	EP 1.1: Adicionar una nueva estructura introduciendo datos validos	Se insertan todos los datos. Se presiona el botón Guardar. Se muestra mensaje estructura guardada.
		Ep1.2: Adicionar una nueva estructura introduciendo datos en blanco	Se introducen datos en blanco en campos requeridos. Se presiona en el botón Guardar. Se muestra mensaje Este Campo es obligatorio en el campo en blanco.

Tabla 5 Prueba 4

3.3.3 Resultados de las pruebas

Después de haber realizado el proceso de pruebas de aceptación, donde estuvieron presentes tanto el encargado como el cliente se obtuvieron resultados satisfactorios, pues se detectaron algunos errores, que fueron solucionados, pues su ocurrencia impedía el total cumplimiento de los requisitos funcionales definidos por el cliente.

Además, después de haberse realizado las pruebas de caja negra a cada uno de los requisitos se obtuvo como resultados un total de 11 no conformidades en la primera iteración, las cuales fueron resueltas, logrando que en la segunda iteración no existiera ninguna. Todo esto llevo a que el software obtenido tenga todas sus funcionalidades de acuerdo a las especificaciones del cliente y que además cumpla con los requerimientos de rendimiento.

3.4 Conclusiones parciales

Después de haberse realizados las pruebas utilizando las técnicas anteriormente especificadas, se llegaron a las siguientes conclusiones:

•Se demostró el buen funcionamiento del sistema y el cumplimiento de los requerimientos del cliente.

• El cliente confirmó que la aplicación agiliza la gestión de roles de las aplicaciones de la empresa.

• Se confirmó que se requieren varias iteraciones de pruebas de caja negra

para garantizar que el producto final cumpla con las funcionalidades exigidas por el cliente, aunque en este caso solo fueron necesario 2 iteraciones.

Conclusiones Generales

A lo largo de toda la investigación se llevaron a cabo un conjunto de tareas, arrojando los siguientes resultados:

- Se describió el estado del arte y se confeccionó un marco teórico referencial después de haberse realizado un estudio acerca de las herramientas y tecnologías a utilizar, determinando así, se encuentran: contar con una herramienta que permita gestionar de forma integrada y segura la información de módulos y funcionalidades, y los roles del negocio de cada una de las aplicaciones, así como la gestión de acceso de los usuarios mediante la asignación de roles y entidades.
- Se realizó el modelado del negocio teniendo en cuenta los pasos que describe ProDeSoft para su realización, entre ellos, la especificación de los requisitos funcionales, el diseño de la base de datos y los diagramas de actividades, diagrama de secuencia, logrando una mayor comprensión del negocio.
- Se implementó el sistema cumpliendo con las políticas y los estándares de desarrollo definidos por ProDeSoft y se realizaron pruebas de validación que permitieron obtener una versión estable del sistema.

Referencias Bibliográficas

APACHE, 2023. Welcome! - The Apache HTTP Server Project. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://httpd.apache.org/.

APACHEFRIENDS, 2023. Download XAMPP. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://www.apachefriends.org/es/download.html.

API MANAGER, 2023. API Manager - On-Premise and in the Cloud. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://wso2.com/es/api-manager/.

(BAC, 2023. ¿Qué es el control de acceso basado en roles (RBAC)? *Cloudflare* [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://www.cloudflare.com/es-es/learning/access-management/role-based-access-control-rbac/.

BUILDER PATTERN, 2023. Builder Pattern: soluciones de software más rápidas con el patrón Builder - IONOS. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/patron-de-diseno-builder/.

DOCTRINE, 2023. Doctrine: PHP Open Source Project. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://www.doctrine-project.org/index.html.

FREIXAS, Y. y NOA, V., 2015. *Sistema para el control de acceso a los laboratorios del Centro de Tecnologías para la Formación desarrollo*. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de Ciencias Informátcas.

GLOSSAIRE, 2023. Modelo conceptual de datos | Glossaire eau, milieu marin et biodiversité. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://glossaire.eauetbiodiversite.fr/es/concept/modelo-conceptual-de-datos.

ISO/IEC 27000, 2018. ISO/IEC 27000:2018. *ISO* [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://www.iso.org/standard/73906.html.

JETBRAINS, 2023. PhpStorm: IDE de PHP y editor de código de JetBrains. *JetBrains* [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://www.jetbrains.com/es-es/phpstorm/.

KEEPCODING, R., 2023. ¿Qué es el Control de Acceso Discrecional? [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://keepcoding.io/blog/que-es-el-control-de-acceso-discrecional/.

MAC, 2023. Mandatory access control (MAC): ¿cómo funciona? - IONOS. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://www.ionos.es/digitalguide/servidores/seguridad/que-es-el-mandatory-access-control-mac/.

MARTÍNEZ, M., 2023. Qué son los Patrones de Diseño de software / Design Patterns. *Profile Software Services* [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://profile.es/blog/patrones-de-diseno-de-software/.

MICROSOFT, 2023a. Control de acceso basado en roles de Azure (RBAC de Azure) frente a las directivas de acceso. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://learn.microsoft.com/es-es/azure/key-vault/general/rbac-access-policy.

MICROSOFT, 2023b. ¿Qué es Access Control? |. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://www.microsoft.com/es/security/business/security-101/what-is-access-control.

MVC, 2023. MVC - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web | MDN. [en línea]. [consulta: 3 septiembre 2023]. Disponible en: https://developer.mozilla.org/es/docs/Glossary/MVC.

OMG, 2023. Qué es el lenguaje unificado de modelado (UML). *Lucidchart* [en línea]. [consulta: 5 diciembre 2023]. Disponible en:

https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml.

PÉREZ, M., 2016. *Sistema de control de acceso automatizado para los laboratorios de la Universidad de las Ciencias Informáticas.* rabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas.

PGADMIN, 2023. pgAdmin - PostgreSQL Tools. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://www.pgadmin.org/.

PHP: Hypertext Preprocessor. [en línea], 2023. [consulta: 5 diciembre 2023]. Disponible en: https://www.php.net/.

POSTGRESQL, 2023. PostgreSQL: Upcoming Events. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://www.postgresql.org/about/events/.

PRODESOFT, 2012. *Proceso de Desarrollo y Gestión de Proyectos de Software.* 2012. S.l.: XETID.

REFACTORING.GURU, 2023. Singleton. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://refactoring.guru/es/design-patterns/singleton.

RIVAS, M., 2016. *IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE ACCESO PARA MEJORAR LA SEGURIDAD DE LA INFORMACIÓN DE LA EMPRESA SNX S.A.C.* Tesina para optar el Título Profesional de Ingeniero de Sistemas. S.l.: UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS.

SYMFONY, 2023. Symfony, High Performance PHP Framework for Web Development. [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://symfony.com/releases/6.0.

TÉLLEZ, D. y GUEVARA, R., 2015. *Componente para facilitar el proceso de autenticación de usuarios en aplicaciones informáticas en instituciones de salud.* Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. Ciudad de la Habana: Universidad de Ciencias Informáticas.

TOUNIS, Y., KIFAYAT, K. y MERABTI, M., 2014. An access control model for cloud computing. *Journal of Information Security and Application 1*, vol. 9, no. 1,

VEGA, E., 2021. *SEGURIDAD DE LA INFORMACIÓN* [en línea]. España: Editorial Área de Innovación y Desarrollo,S.L. Disponible en: https://doi.org/10.17993/tics.2021.4.

VISUAL PARADIGM, 2023. What's New in Visual Paradigm? [en línea]. [consulta: 5 diciembre 2023]. Disponible en: https://www.visual-paradigm.com/whats-new/.

WANG, H., GUO, X., FAN, Y. y BI, J., 2014. Extended access control and recommendation methods for enterprise knowledge management system. *IERI Procedia*, vol. 10,

YANG, J., SI, Z., ZHEN, H., MU, L., JING, N. y PING, N., 2013. Access control for rural medical and health collaborative working platform. *The Journal of Chine Universities of Posts and Telecommunication*, vol. 20,

Anexos

Mapa Conceptual


Sistema de Gestión de Roles
Gestionar Perfiles de usuario de aplicaciones
Ésta funcionalidad permite listar los usuarios que se ha registrados en las apicaciones previamente registrados. Modificar acceso a otras aplicaciones, Cambiar rol y estructura para cada una de las aplicaciones
Gestionar los roles de las aplicaciones
Desde esta funcionalidad se permite listar, crear, modificar y eliminar los roles del negocio que conforman cada aplicación que se ha desarrollado y que están registrada mediante la funcionalidad anterior, en el sistema. También se permite hacer la asociación roles-funcionalidades
Configurar Sistema
o Gestionar los usuarios del sistema
Podrán tener roles de Desarrollador, administrador y Gestor de la Configuración. Estos roles no son gestionables y estos usuarios serán los encargados de ejecutar las funcionalidades asociadas a las aplicaciones
o Gestionar Estructura y composición
Desde esta funcionalidad se podrá listar, adicionar, modificar y eliminar la información asociada al catálogo o nomenclador nacional de entidades del país
Gestionar la información las aplicaciones
Desde esta funcionalidad se permite listar, adicionar, modificar y eliminar la información de los módulos o subsistemas que componen una aplicación informática. Así como la información de las funcionalidades que forman cada uno de estos módulos o subsistemas.