

Desarrollo de Componentes Centrales para el Módulo de Configuración General del Sistema de Información Administrativa.

Isnardo Corredor Ariza

Práctica empresarial presentada para optar al título de Ingeniero de Sistemas

Director:

Gabriel Rodrigo Pedraza Ferreira  
Doctor en Ciencias de la Computación.

Tutor:

Jacksson Sonny González Bayona  
Magíster en Administración de Organizaciones.

Universidad Industrial de Santander  
Facultad de Ingeniería Fisicomecánicas  
Escuela de Ingeniería de Sistemas  
Bucaramanga

2023

### **Agradecimientos**

A mi familia, quienes desde el inicio me brindaron su amor incondicional y apoyo en la decisión de seguir mi carrera universitaria. Su constante aliento y sacrificio hicieron posible este logro.

Agradezco profundamente al profesor Gabriel Rodrigo Pedraza, por su invaluable guía y compromiso como director de este proyecto. Su conocimiento y orientación fueron fundamentales para el desarrollo de este trabajo.

Mi gratitud también se extiende al Magister Jacksson González, cuya colaboración en la práctica contribuyó significativamente a mi formación profesional y al enriquecimiento de esta investigación.

Agradezco al profesor Gilberto Díaz por incluirme en el proyecto de Renovación de los Sistemas de Información (RSI). Su confianza en mí me motivó a dar lo mejor de mí en este proceso.

A mis amigos, Neygel Prada, Nathalia Ocampo, Jorge Cárdenas, Daniel Torres, Juan Camilo Gómez y Mauricio Romero, quienes han sido pilares fundamentales en mi camino. Gracias por su apoyo incondicional, por alentarme a no rendirme y por compartir momentos inolvidables que quedan grabados en mi corazón.

A cada uno de ustedes, gracias por creer en mí y por ser parte de este camino lleno de aprendizaje y crecimiento. Esta tesis lleva también un pedacito de cada uno de ustedes.

**Contenido**

	<b>Pág.</b>
Introducción .....	13
1. Generalidades.....	15
1.1 Planteamiento del Problema .....	15
2. Objetivos .....	17
2.1 Objetivo General.....	17
2.2 Objetivos Específicos.....	17
3. Metodología .....	18
3.1 Metodología a Implementar .....	18
3.1.1 Fases del Proyecto.....	18
3.1.1.1 Definición de Proyecto y Análisis de Viabilidad.....	18
3.1.1.2 Planificación. ....	19
3.1.1.3 Ejecución.....	19
3.1.1.4 Seguimiento y Control. ....	20
3.1.1.5 Clausura y Evaluación. ....	20
3.2 Alcance .....	20
3.3 Ambientes de Desarrollo.....	21
3.3.1 PREDEV.....	22
3.3.2 DEVELOP .....	22
3.3.3 Preproducción .....	22
3.3.4 Producción .....	22

DESARROLLO DE COMPONENTES CENTRALES	4
4. Marco de Referencia .....	23
4.1 Tecnologías Actuales .....	23
4.1.1 INFORMIX.....	23
4.1.2 JAVA .....	23
4.1.3 .NET.....	23
4.2 Fundamentos Teóricos .....	23
4.2.1 DEVOPS .....	24
4.2.2 CI/CD (Integración Continua/Distribución Continua).....	24
4.2.3 Arquitectura Basada en Microservicios .....	25
4.2.4 API REST .....	26
4.2.5 Control De Versiones.....	26
4.2.6 Marcos Ágiles .....	27
4.2.7 JWT (JSON WEB TOKEN) .....	27
4.2.8 Requerimientos Funcionales y No Funcionales.....	28
4.2.9 Actores .....	28
4.2.10 Casos de Uso.....	28
4.2.11 Diagramas de Secuencia .....	28
4.2.12 Diagramas de Actividad.....	29
4.2.13 Modelo Entidad-Relación .....	29
4.2.14 Arquitectura de Software .....	29
5. Requerimientos .....	30
5.1 Análisis de Requerimientos y Diseño de Funcionalidades en el Desarrollo de Tareas	
Asignadas .....	30

DESARROLLO DE COMPONENTES CENTRALES	5
5.1.1 Home del Módulo Configuración General .....	30
5.1.2 Acción de Generar Nueva Contraseña en el Listado de Usuarios .....	33
5.1.3 Pintar Menús de Acceso Directo en la Barra Superior Derecha .....	36
5.1.4 Zonificar el Menú de Navegación Lateral Izquierdo.....	39
5.1.5 Buscador en el Menú de Tercer Nivel .....	43
5.1.6 Estabilización de Test de Frontend .....	47
5.1.7 CRUD de Autoregistros.....	53
6. Diseño .....	58
6.1 Arquitectura de la Renovación de los Sistemas de Información (RSI) .....	58
6.2 Tecnologías Implementadas.....	59
6.2.1 Angular .....	59
6.2.2 JAVA SPRINGBOOT .....	60
6.2.3 ORACLE DB.....	60
6.2.4 GITLAB.....	60
6.2.5 TAIGA (Gestión de Proyecto).....	61
6.3 Proceso de Desarrollo con GITLAB, KONG Y CI/CD.....	61
6.4 Modelos Entidad Relación (ER) .....	62
6.4.1 Modelo ER HOME de Configuración General.....	62
6.4.2 Modelo ER de Acción de Generar Nueva Contraseña.....	63
6.4.3 Modelo ER de Pintar Menús de Acceso Directo en la Barra Superior Derecha.....	64
6.4.4 Modelo ER de Zonificar el Menú de Navegación Lateral Izquierdo.....	65
6.4.5 Modelo ER de Buscador de Menú de Tercer Nivel.....	67
6.4.6 Modelo ER de Estabilización de Test de Frontend.....	68

DESARROLLO DE COMPONENTES CENTRALES	6
6.4.7 Modelo ER de Crud de Autoregistros.....	68
7. Implementación.....	71
7.1 Herramientas y Procedimientos para el Desarrollo del Proyecto .....	71
7.2 Subir Código al Repositorio .....	73
7.3 Componentes Funcionales .....	77
7.3.1 Home de la Configuración General .....	77
7.3.2 Acción de Generar Contraseña en el Listado de Usuarios.....	77
7.3.3 Pintar Menús de Acceso Directo en la Barra Superior Derecha .....	78
7.3.4 Zonificar el Menú de Navegación Lateral Izquierdo.....	79
7.3.5 Buscador en el Menú de Tercer Nivel .....	80
7.3.6 Estabilización de Test de Frontend .....	81
7.3.7 Crud de Autoregistros .....	83
8. Validación.....	85
8.1 Pruebas para la Home de Configuración general.....	87
8.2 Pruebas para Acción de Generar Nueva Contraseña en Listado de Usuarios.....	91
8.3 Pruebas para Pintar Menús de Acceso Directo en la Barra Superior Derecha .....	98
8.4 Pruebas para Zonificar el Menú de Navegación Vertical Izquierdo.....	113
8.5 Pruebas para el Buscador en el Menú de Tercer Nivel.....	119
8.6 Pruebas para Estabilización de Test de Frontend .....	122
8.7 Pruebas para Crud de Autoregistros .....	125
9. Conclusiones .....	135
10. Trabajo a Futuro.....	137
Referencias.....	139

**Lista de Figuras**

	<b>Pág.</b>
Figura 1. <i>Explicación gráfica de CI/CD.</i> .....	25
Figura 2. <i>Diagrama de una arquitectura de microservicios.</i> .....	26
Figura 3. <i>Diagrama de versiones distribuido en ramas.</i> .....	27
Figura 4. <i>Diagrama de secuencia UML.</i> .....	29
Figura 5. <i>Diagrama de casos de uso de la Home de configuración general.</i> .....	31
Figura 6. <i>Diagrama de secuencias de la HOME de configuración general.</i> .....	32
Figura 7. <i>Diagrama de actividades de la HOME de configuración general.</i> .....	32
Figura 8. <i>Diagrama de casos de uso de la acción de generar nueva contraseña</i> .....	34
Figura 9. <i>Diagrama de secuencias de acción de generar nueva contraseña.</i> .....	35
Figura 10. <i>Diagrama de actividades de acción de generar nueva contraseña.</i> .....	36
Figura 11. <i>Diagrama de casos de uso de pintar menús de acceso directo.</i> .....	38
Figura 12. <i>Diagrama de actividades de pintar menús de acceso directo.</i> .....	38
Figura 13. <i>Diagrama de actividades de pintar menús de acceso directo.</i> .....	39
Figura 14. <i>Diagrama de casos de uso sobre la zonificación del menú lateral izquierdo.</i> .....	41
Figura 15. <i>Diagrama de secuencias sobre la zonificación del menú lateral izquierdo.</i> .....	42
Figura 16. <i>Diagrama de actividades sobre la zonificación del menú lateral izquierdo.</i> .....	43
Figura 17. <i>Diagrama de casos de uso del buscador de menú de tercer nivel.</i> .....	45
Figura 18. <i>Diagrama de secuencias del buscador de menús.</i> .....	46
Figura 19. <i>Diagrama de actividades del buscador de menús.</i> .....	47
Figura 20. <i>Salida de consola usando comando ng test.</i> .....	50

DESARROLLO DE COMPONENTES CENTRALES	8
Figura 21. Salida de consola en el navegador .....	50
Figura 22. Archivo karma.conf.js del proyecto de CORE.....	51
Figura 23. Diagrama de secuencias de estabilización de test de frontend. ....	52
Figura 24. Diagrama de actividades de estabilización de test de frontend. ....	53
Figura 25. Diagrama de casos de uso de CRUD de autoregistro. ....	55
Figura 26. Diagrama de secuencia de CRUD de autoregistro. ....	56
Figura 27. Diagrama de actividades de CRUD de autoregistro. ....	57
Figura 28. Tecnologías utilizadas en RSI. ....	58
Figura 29. Diagrama Entidad Relación de la tabla Usuarios y Personas. ....	63
Figura 30. Diagrama entidad-relación de las rutas navegación. ....	64
Figura 31. Diagrama entidad-relación de las rutas navegación. ....	65
Figura 32. Diagrama entidad-relación de los roles rutas navegación. ....	65
Figura 33. Diagrama entidad-relación de los usuarios rutas navegación. ....	66
Figura 34. Diagrama entidad-relación de los usuarios roles rutas navegación. ....	67
Figura 35. Diagrama Entidad-Relación de autoregistro. ....	68
Figura 36. Diagrama Entidad-Relación de autoregistro y Roles. ....	69
Figura 37. Diagrama Entidad-Relación de autoregistro y usuarios. ....	69
Figura 38. Tecnologías utilizadas en RSI. ....	73
Figura 39. Interfaz de Gitkraken Client. ....	76
Figura 40. Home del módulo de configuración general .....	77
Figura 41. Interfaz del buscador de usuarios. ....	77
Figura 42. Modal de la acción de generar nueva contraseña. ....	78
Figura 43. Toogle para habilitar el acceso directo de un menú. ....	78

DESARROLLO DE COMPONENTES CENTRALES	9
Figura 44. <i>Íconos de menús de acceso directo.</i> .....	79
Figura 45. <i>Zonificación del menú lateral izquierdo.</i> .....	79
Figura 46. <i>Input del buscador de menús.</i> .....	80
Figura 47. <i>Buscador de menús en función.</i> .....	80
Figura 48. <i>Salida de Consola del módulo CORE.</i> .....	81
Figura 49. <i>Salida de consola en el navegador del módulo de CORE.</i> .....	81
Figura 50. <i>Salida de Consola del módulo AUTHORIZATION.</i> .....	82
Figura 51. <i>Salida de consola en el navegador del módulo de AUTHORIZATION.</i> .....	82
Figura 52. <i>CRUD de autoregistro.</i> .....	83
Figura 53. <i>Crear autoregistro.</i> .....	83
Figura 54. <i>Editar autoregistro.</i> .....	84
Figura 55. <i>Eliminar autoregistro.</i> .....	84

**Lista de Tablas**

	<b>Pág.</b>
Tabla 1. <i>Pruebas para la Home de Configuración general - Casos de prueba</i> .....	87
Tabla 2. <i>Pruebas para la Home de Configuración general - Lista de chequeo</i> .....	90
Tabla 3. <i>Pruebas para Acción de Generar Nueva Contraseña en Listado de Usuarios - Casos de prueba</i> .....	91
Tabla 4. <i>Pruebas para Acción de Generar Nueva Contraseña en Listado de Usuarios - Lista de chequeo</i> .....	97
Tabla 5. <i>Pruebas para Pintar Menús de Acceso Directo en la Barra Superior Derecha - Casos de prueba</i> .....	98
Tabla 6. <i>Pruebas para Pintar Menús de Acceso Directo en la Barra Superior Derecha - Lista de chequeo</i> .....	110
Tabla 7. <i>Pruebas para Zonificar el Menú de Navegación Vertical Izquierdo - Casos de prueba</i> .....	113
Tabla 8. <i>Pruebas para Zonificar el Menú de Navegación Vertical Izquierdo - Lista de chequeo</i> .....	118
Tabla 9. <i>Pruebas para el Buscador en el Menú de Tercer Nivel - Casos de prueba</i> .....	119
Tabla 10. <i>Pruebas para el Buscador en el Menú de Tercer Nivel - Lista de chequeo</i> .....	121
Tabla 11. <i>Pruebas para Estabilización de Test de Frontend - Casos de prueba</i> .....	122
Tabla 12. <i>Pruebas para Crud de Autoregistros - Casos de prueba</i> .....	125
Tabla 13. <i>Pruebas para Crud de Autoregistros - Lista de chequeo</i> .....	130

## Resumen

**Título:** Desarrollo de Componentes Centrales para el Módulo de Configuración General del Sistema de Información Administrativa\*.

**Autor:** Isnardo Corredor Ariza\*\*

**Palabras clave:** Renovación, arquitectura de microservicios, marcos ágiles, GitLab, pruebas de calidad, experiencia del usuario

### Descripción

El proyecto aborda la renovación del sistema de información de una universidad, cuyo módulo central es la "Configuración General". Se plantea la necesidad de mantener los sistemas actualizados y alineados con las últimas tendencias tecnológicas.

Los objetivos se centran en implementar una arquitectura de microservicios, utilizar marcos ágiles y prácticas DevOps para agilizar el desarrollo y entrega continua. Se busca mejorar la experiencia del usuario, eficiencia y seguridad del sistema.

El desarrollo del proyecto se basa en la creación de microservicios independientes para cada funcionalidad del sistema, con tecnologías como Angular y Spring Boot. Se adopta un enfoque de colaboración entre desarrolladores y administradores, utilizando GitLab como controlador de versiones para el desarrollo y pruebas.

Las validaciones se realizan mediante pruebas de calidad, DevTest con compañeros desarrolladores y pruebas de funcionamiento con el personal administrativo encargado de la gestión de talento humano de la universidad.

En conclusión, el proyecto logra una renovación efectiva del sistema de información mediante una arquitectura moderna y métodos ágiles, lo que asegura una mayor flexibilidad, eficiencia y mejora en la experiencia del usuario final.

---

\* Trabajo de Grado

\*\* Facultad de Ingeniería Fisicomecánicas. Escuela de Ingeniería de Sistemas. Director: Gabriel Rodrigo Pedraza Ferreira, Doctor en Ciencias de la Computación. Tutor: Jacksson Sonny González Bayona, Magíster en Administración de Organizaciones.

**Abstract**

**Title:** Development of Central Components for the General Configuration Module of the Administrative Information System\*.

**Author:** Isnardo Corredor Ariza\*\*

**Keywords:** Renovation, microservices architecture, agile frameworks, GitLab, quality testing, user experience

**Description**

The project addresses the renovation of a university's information system, with the central module being "General Configuration." The need to keep the systems up-to-date and aligned with the latest technological trends is identified.

The objectives focus on implementing a microservices architecture, using agile frameworks and DevOps practices to streamline development and enable continuous delivery. The aim is to enhance user experience, system efficiency, and security.

The project's development is based on creating independent microservices for each system functionality, using technologies such as Angular and Spring Boot. A collaborative approach between developers and administrators is adopted, utilizing GitLab as the version control system for development and testing.

Validation is carried out through quality tests, DevTest with fellow developers, and functional testing with administrative staff responsible for human resources management at the university.

In conclusion, the project achieves an effective information system renovation through a modern architecture and agile methodologies, ensuring greater flexibility, efficiency, and an improved end-user experience.

---

\* Degree work

\*\* Faculty of Physical-Mechanical Engineering. School of Systems Engineering. Director: Gabriel Rodrigo Pedraza Ferreira, PhD in Computer Science. Tutor: Jacksson Sonny González Bayona, Master in Organization Administration.

## **Introducción**

Un sistema de información (SI) se refiere a un conjunto de mecanismos y datos complejos que tienen como objetivo administrar y procesar información de manera eficiente y accesible. Estos sistemas son fundamentales para la toma de decisiones en las organizaciones, ya que convierten datos en información útil y significativa para respaldar las operaciones y estrategias empresariales.

Configurar un sistema de información es una tarea aún más compleja, ya que implica diseñar y ajustar los componentes del sistema para satisfacer las necesidades específicas de una organización. Requiere la configuración adecuada de los recursos, la integración de los procesos y la adaptación de las funcionalidades para lograr un sistema que sea eficaz y eficiente.

A medida que los sistemas de información crecen en tamaño y complejidad, se vuelve necesario adoptar enfoques más avanzados como el uso de microservicios. Los microservicios son una arquitectura de desarrollo de software que permite dividir una aplicación en componentes más pequeños e independientes, conocidos como microservicios. Estos microservicios pueden ser desarrollados, implementados y escalados de forma individual, lo que facilita el mantenimiento y la evolución de sistemas de información grandes y complejos. El sistema de información administrativa (SIA) de la gestión de Talento Humano actual de la Universidad Industrial de Santander desempeña un papel crucial en el funcionamiento eficiente de la institución. Es fundamental mantenerlo actualizado y en constante evolución para adaptarse a los cambios tecnológicos y a las necesidades en constante cambio de la comunidad universitaria. Un sistema de información actualizado garantiza la disponibilidad de información precisa y oportuna, promueve la eficiencia en los procesos administrativos y académicos, y proporciona una base

sólida para la toma de decisiones estratégicas. Por lo tanto, la renovación y mejora continua del sistema de información son elementos clave para el éxito de la Universidad Industrial de Santander en el contexto actual de rápido avance tecnológico.

## 1. Generalidades

### 1.1 Planteamiento del Problema

Los sistemas de información utilizados por el personal de la Universidad Industrial de Santander, cuyo desarrollo se inició en el año 2000, se basaron en tecnologías de vanguardia de ese entonces, como ".NET", JAVA 5, Informix, entre otras. Sin embargo, las versiones utilizadas en esta implementación eran poco conocidas y contaban con escasa documentación, lo cual dificultaba brindar un soporte adecuado al sistema. Este hecho constituye uno de los principales motivos que impulsan la renovación de los sistemas de información, mediante el empleo de técnicas más modernas y ampliamente documentadas, que gozan de una amplia aplicación en el mundo actual. De esta manera, se logrará establecer una base sólida y permitir futuras actualizaciones al sistema, prolongando así su vida útil.

El sistema de información (SI) actual utilizado para gestionar el talento humano en la Universidad Industrial de Santander es efectivo y permite llevar a cabo los procesos y actividades necesarios en los diferentes módulos. Sin embargo, presenta deficiencias en términos de eficiencia. Esto se debe principalmente a la existencia de diferentes arquitecturas de software, al uso de tecnologías obsoletas, a retrasos en los tiempos de respuesta y a una interacción deficiente con el usuario.

La problemática a abordar en este proyecto se centra en la renovación del componente encargado de la configuración de la plataforma, específicamente en el módulo de "CONFIGURACIÓN GENERAL". Este componente permitirá parametrizar y configurar el Sistema de Información Administrativo (SIA) de manera más eficiente y efectiva. Para lograrlo, se emplearán funciones transversales que facilitarán la configuración de la plataforma en su

totalidad. Esta iniciativa representa un pequeño aporte al gran proyecto de RENOVACIÓN DE LOS SISTEMAS DE INFORMACIÓN (RSI) revisado y avalado por la Universidad Industrial de Santander, bajo la dirección del señor rector HERNÁN PORRAS DÍAZ.

## 2. Objetivos

### 2.1 Objetivo General

Realizar el análisis y desarrollo de componentes y nuevas funcionalidades del módulo de configuración general del sistema en la división de la gestión de talento humano, haciendo uso de *frameworks* y prácticas de programación, como patrones de diseño, para avalar la usabilidad, escalabilidad, robustez y soporte de los componentes del sistema de información administrativa (SIA).

### 2.2 Objetivos Específicos

Analizar los requerimientos funcionales, no funcionales y casos de uso para los componentes pertenecientes al módulo de configuración general del sistema de información administrativa (SIA).

Implementar las funcionalidades transversales por medio de *frameworks* para los módulos referidos en base a diseños propuestos para el sistema de información administrativa (SIA).

Desarrollar los componentes relacionados con la administración de usuarios, roles, tipos de roles y menús del sistema de información administrativa (SIA).

Evaluar las funcionalidades y componentes desarrollados, basado en estándares de usabilidad y diseño orientados para una mayor facilidad de uso por parte del usuario.

### 3. Metodología

#### 3.1 Metodología a Implementar

El análisis, desarrollo e implementación del proyecto es realizado bajo el marco ágil Scrum, la cual es un marco que permite el trabajo colaborativo entre equipos. El autor de este documento será el encargado la mayor parte del desarrollo *frontend* y *backend* del plan. Además, las demás partes involucradas en la realización de la planificación, tales como apartados visuales, el liderazgo del equipo y microservicios, fueron asignados por el equipo de Renovación de Sistemas de Información (RSI) de la Universidad Industrial de Santander.

##### 3.1.1 Fases del Proyecto

El desarrollo del Sistema de Información Administrativa (SIA) se basó en 5 fases, las cuales fueron apoyadas en el uso del marco ágil SCRUM. De esta manera, el proyecto tiene un mejor control gracias a las siguientes fases:

**3.1.1.1 Definición de Proyecto y Análisis de Viabilidad.** En el proceso de definición de proyecto y análisis de viabilidad del problema a resolver, se han definido la solución, los objetivos, el alcance, las limitaciones, el marco teórico, la factibilidad y el cronograma del proyecto. Para lograr esta definición, se llevaron a cabo diversas actividades y análisis detallados.

- Planteamiento del problema.
- Definición de objetivos.
- Definición de alcances y limitaciones.
- Desarrollo del marco teórico.
- Desarrollo de las factibilidades del proyecto.
- Desarrollo del cronograma del proyecto.

**3.1.1.2 Planificación.** En la etapa de planificación, se definen los requerimientos funcionales y no funcionales, los actores, los casos de uso, los diagramas de secuencia, los diagramas de actividades, los modelos entidad-relación y la arquitectura base del proyecto. Estas actividades son fundamentales para establecer una sólida base de trabajo y guiar el desarrollo del proyecto de manera eficiente:

- Definición de nuevos requerimientos funcionales y no funcionales.
- Definición de actores.
- Definición de casos de uso.
- Definición diagramas de secuencia.
- Definición diagramas de actividad.
- Definición modelo entidad-relación.
- Arquitectura de software.

**3.1.1.3 Ejecución.** Durante la etapa de ejecución, se comienza con la configuración de la arquitectura del proyecto y se procede a la codificación de los componentes, aplicando buenas prácticas de desarrollo, manteniendo un código limpio, según los principios del *clean code* y bien documentado:

- Desarrollo de la arquitectura principal del proyecto.
- Desarrollo de la *HOME* de configuración general.
- Desarrollo de la acción de generar nueva contraseña en el listado de usuarios.
- Desarrollo de pintar menús de acceso directo en la barra superior derecha.
- Desarrollo de zonificar el menú de navegación vertical izquierdo.
- Desarrollo de buscador en el menú de tercer nivel.
- Desarrollo de estabilización de test de *frontend*.

- Desarrollo de CRUD de autoregistros.

Este ciclo se ejecuta en alternancia con el período de seguimiento y control.

**3.1.1.4 Seguimiento y Control.** Durante la fase de seguimiento y control, se llevan a cabo pruebas de calidad de los componentes desarrollados, enfocadas en los usuarios finales. Las actividades programadas incluyen pruebas para las siguientes actividades:

- Pruebas para la *HOME* de configuración general.
- Pruebas para acción de generar nueva contraseña en el listado de usuarios.
- Pruebas para pintar menús de acceso directo en la barra superior derecha.
- Pruebas para el menú de navegación vertical izquierdo.
- Pruebas para buscador en el menú de tercer nivel.
- Pruebas para estabilización de test de *frontend*.
- Pruebas para CRUD de autoregistros.
- Subir código al repositorio.

Este período se ejerce en alternancia con la fase de ejecución.

**3.1.1.5 Clausura y Evaluación.** Se entregan los componentes desarrollados y con las pruebas realizadas al cliente en ambiente DEVELOP. Para la fase de clausura y evaluación se desarrollaron las siguientes actividades:

- Entrega de componentes funcionales.

## 3.2 Alcance

El proyecto se enfoca en satisfacer las necesidades del nuevo sistema de información administrativa (SIA), priorizando el módulo de Configuración General. Este módulo permitirá gestionar la configuración de la plataforma, parametrizando el SIA en diferentes módulos de manera transversal y específica.

En el marco de este proyecto, se desarrollaron los siguientes flujos de trabajo:

- **HOME de Configuración General:** Proporcionar una interfaz principal desde donde se podrán acceder y gestionar las diferentes funcionalidades de configuración general.
- **Acción de generar nueva contraseña:** Permitir a los usuarios generar una nueva contraseña para su cuenta, siguiendo los protocolos de seguridad establecidos.
- **Pintar menús de acceso directo en la barra superior derecha:** Desplegar iconos de menús en la barra superior derecha de la plataforma, permitiendo una navegación ágil entre diferentes menús sin perder la ubicación actual.
- **Zonificar el menú lateral izquierdo:** Estructurar el menú lateral izquierdo para visualizar los permisos de navegación asociados a roles y usuarios, facilitando la administración de los accesos.
- **Buscador en el menú de tercer nivel:** Implementar un buscador que permitirá filtrar y encontrar rápidamente las opciones dentro del menú de tercer nivel, agilizando la navegación del usuario.
- **Estabilización de test de frontend:** Realizar acciones y mejoras para garantizar la estabilidad y confiabilidad de las pruebas unitarias de *frontend*.
- **CRUD de autoregistros:** Desarrollar las operaciones básicas (crear, leer, actualizar, eliminar) para gestionar los autoregistros en la plataforma.

### 3.3 Ambientes de Desarrollo

El proyecto consta de 4 ambientes de desarrollo y despliegue.

### **3.3.1 PREDEV.**

El ambiente "PREDEV" es utilizado para el desarrollo de componentes, pruebas iniciales, revisión de calidad del código, documentación, diseño y funcionamiento. Estas pruebas, llamadas DevTest, son ejecutadas por compañeros desarrolladores del equipo.

### **3.3.2 DEVELOP**

En este ambiente se ejecutan las pruebas de calidad por el equipo de QA, revisando el diseño requerido y funcionamiento de la historia de usuario, dicho entorno está orientado a usuarios finales.

### **3.3.3 Preproducción**

El ambiente "PREPRODUCCIÓN" es donde el personal administrativo de gestión de talento humano realiza la primera toma de contacto con los componentes, ejecuta pruebas de funcionamiento y verifica el flujo con datos reales.

### **3.3.4 Producción**

El ambiente "PRODUCCIÓN" es el marco final de entrega de los componentes, donde se encuentra el aplicativo desarrollado por completo y será utilizado por el personal administrativo de gestión de talento humano de la universidad.

## **4. Marco de Referencia**

### **4.1 Tecnologías Actuales**

Los sistemas actuales utilizados por el personal administrativo encargado de gestionar los recursos humanos en el ente educativo fueron desarrollados utilizando el siguiente paquete de tecnologías:

#### ***4.1.1 INFORMIX***

IBM – Informix es una base de datos diseñada para ser rápida y flexible con una capacidad de integrar sencillamente datos de SQL, NoSQL/JSON y series temporales y espaciales. Además, ocupa poco espacio y posee unas increíbles capacidades de autogestión para ser adecuada a los entornos integrados de gestión de datos. (Informix - Visión general, s. f.).

#### ***4.1.2 JAVA***

Java, fundado en 1995, destaca por su enfoque orientado a objetos y su capacidad de ejecutarse en múltiples sistemas operativos. Ofrece soporte de red integrado y permite la ejecución segura de código de fuentes remotas. Su facilidad de uso lo ha mantenido relevante en la era digital. (Carranza, 2021)

#### ***4.1.3 .NET***

.NET es una plataforma de código abierto que permite crear aplicaciones para diferentes sistemas operativos. Incluye herramientas, bibliotecas y lenguajes para un desarrollo moderno y escalable, respaldado por una comunidad activa de desarrolladores (Emaz, A. y Pérez-Ruiz, 1998).

### **4.2 Fundamentos Teóricos**

En el desarrollo del proyecto, se proporcionarán definiciones fundamentales para comprender las tecnologías, arquitectura y patrones de diseño utilizados. Esto permitirá un entendimiento completo de los conceptos clave involucrados en el proceso de desarrollo.

#### **4.2.1 DEVOPS**

DevOps es una metodología que acerca a los equipos de programación y sistemas mediante nuevas herramientas y prácticas. Esta colaboración aumenta la agilidad y la productividad del negocio (Claranet, 2020).

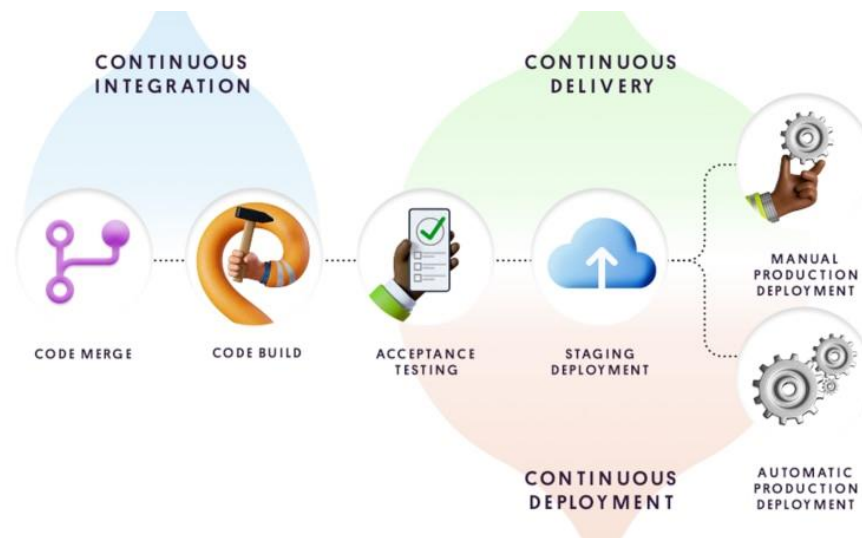
Se trata también, por lo tanto, de un cambio cultural que permite a las empresas acelerar el ciclo de vida de sus aplicaciones (Claranet, 2020).

#### **4.2.2 CI/CD (*Integración Continua/Distribución Continua*)**

CI/CD es una práctica de DevOps que involucra la Integración Continua (CI) y la Entrega Continua/Implementación Continua (CD). La CI implica integrar cambios de código frecuentemente en el repositorio, mientras que la CD automatiza la entrega e implementación de las versiones finales. Las pruebas regulares en CI/CD reducen errores y son esenciales en el flujo de trabajo de DevOps (Technologies, U., s. f.).

**Figura 1.**

*Explicación gráfica de CI/CD.*



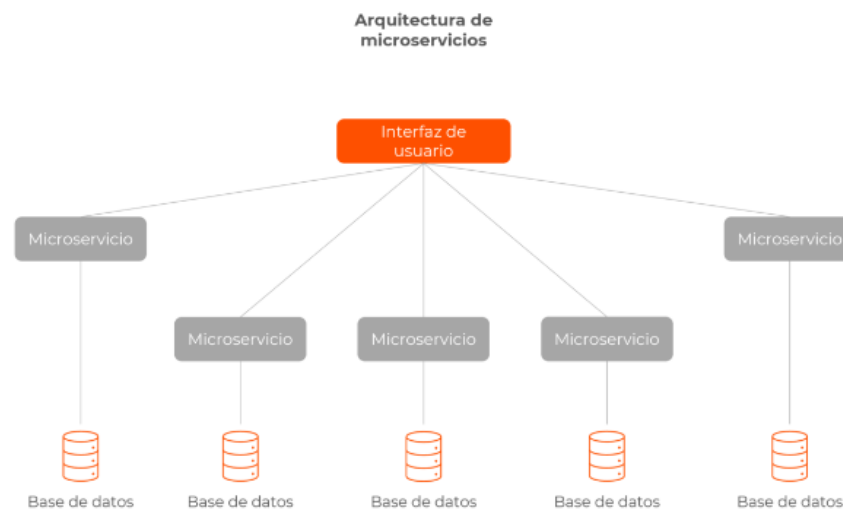
*Nota:* Unity. (sf). *Explicación de CI/CD.* Tomado de: <https://unity.com/es/solutions/what-is-ci-cd>

### ***4.2.3 Arquitectura Basada en Microservicios***

La arquitectura de microservicios es un enfoque de desarrollo de software que consiste en pequeños servicios independientes que se comunican mediante APIs. Proporciona flexibilidad y adaptabilidad al permitir cambios individuales sin afectar al resto del sistema (decide4AI, 2022).

**Figura 2.**

*Diagrama de una arquitectura de microservicios.*



*Nota:* Decide Soluciones. (sf). *Arquitectura de microservicios: qué es, ventajas y desventajas.*

Tomado de: <https://decidesoluciones.es/arquitectura-de-microservicios/>

#### **4.2.4 API REST**

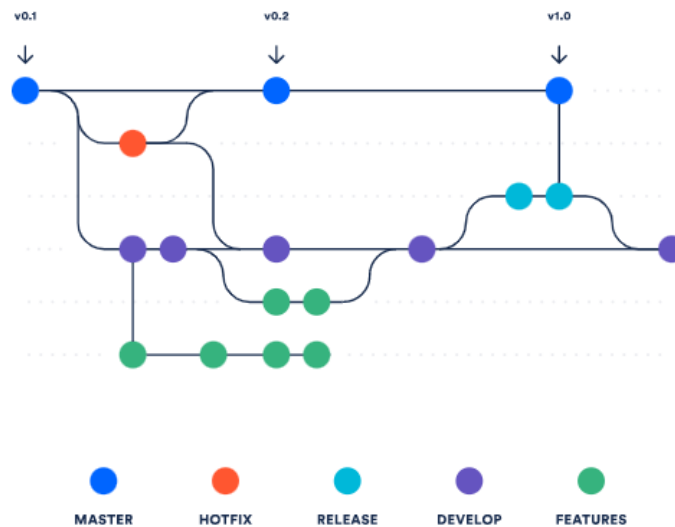
Las API REST son interfaces de programación que facilitan la integración de aplicaciones, especialmente en la arquitectura de microservicios, al seguir los principios de diseño de REST, conocido como transferencia de estado representacional. (IBM Cloud Education, s. f.)

#### **4.2.5 Control De Versiones**

Un sistema de control de versiones (VCS) es una herramienta que gestiona y rastrea cambios en archivos, permitiendo colaboración y control en el desarrollo de software mediante seguimiento de adiciones, eliminaciones y modificaciones (Atlassian, s. f.).

**Figura 3.**

*Diagrama de versiones distribuido en ramas.*



*Nota:* Bitbucket. (sf). *Software de control de versiones para equipos profesionales.* Tomado de: <https://bitbucket.org/product/es/version-control-software>

#### **4.2.6 Marcos Ágiles**

Los marcos ágiles permiten adaptar el trabajo a las condiciones del proyecto, logrando flexibilidad y respuesta inmediata. SCRUM es una metodología ágil que favorece entregas regulares, flexibilidad, productividad e innovación en proyectos complejos y cambiantes. (Sotomayor, S. G., 2022) (Proyectos Ágiles, 2021)

#### **4.2.7 JWT (JSON WEB TOKEN)**

Un JSON Web Token (JWT) es una clave de acceso estandarizada que permite el intercambio seguro de datos entre dos partes. Contiene información importante sobre una entidad, evitando consultas a la base de datos y sesiones en el servidor, lo que lo hace ideal para autenticación y autorización. Los usuarios interactúan indirectamente con el token al ingresar sus credenciales, mientras que la comunicación entre aplicaciones ocurre entre el cliente y el servidor. (Introducción a JSON Web Token (JWT), 2020)

#### ***4.2.8 Requerimientos Funcionales y No Funcionales***

Los requisitos funcionales describen las funciones del sistema y cómo debe responder a comandos para satisfacer las necesidades del usuario. Los requisitos no funcionales, en cambio, establecen limitaciones y restricciones sin afectar la funcionalidad. (Solutions, V., & Jain, A., 2023)

#### ***4.2.9 Actores***

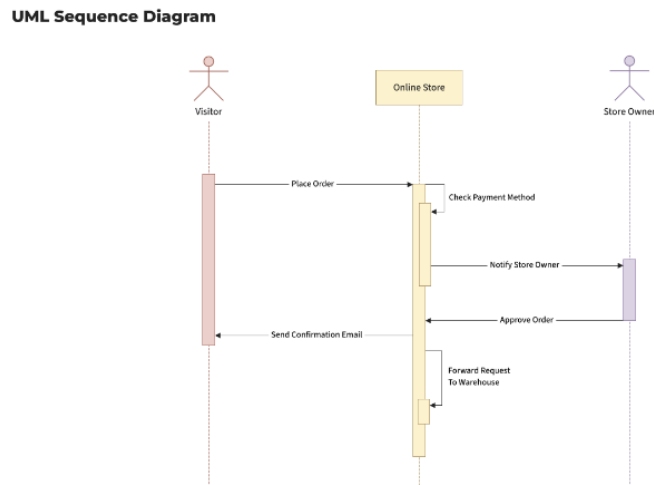
Un actor especifica un papel desempeñado por un usuario o cualquier otro sistema que interactúe con el sujeto. Puede representar roles desempeñados por usuarios humanos, hardware externo u otros sujetos. Los actores están siempre fuera del sistema e interactúan directamente con él iniciando un caso de uso, proporcionándole entradas y/o recibiendo salidas del mismo. Aunque una única instancia física puede desempeñar el papel de varios actores diferentes, un actor no representa necesariamente entidades físicas específicas, por ejemplo, un temporizador que activa el envío de un recordatorio por correo electrónico. (Types of Actor in a Use Case Model, s. f.)

#### ***4.2.10 Casos de Uso***

Los casos de uso son una técnica para capturar requisitos en sistemas de software, proporcionando escenarios de interacción entre el sistema y el usuario para lograr objetivos específicos en un lenguaje cercano al usuario final. (Casos de uso, 2017)

#### ***4.2.11 Diagramas de Secuencia***

Los diagramas de secuencia son herramientas en ingeniería de software para describir interacciones entre objetos en un sistema, mostrando el orden y la ejecución de funcionalidades. (Plantilla de Diagrama de Secuencia UML | Moqups, s. f.)

**Figura 4.** Diagrama de secuencia UML.

*Nota:* Moqups. (sf). *¿Qué es un diagrama de secuencia?* Tomado de: <https://moqups.com/es/templates/diagrams-flowcharts/uml-diagrams/sequence-diagram/>

#### 4.2.12 Diagramas de Actividad

El diagrama de actividad UML muestra el flujo de actividades en un sistema o proceso comercial, permitiendo una visualización detallada del caso de uso y la identificación de necesidades. (Siriwardhana, 2022)

#### 4.2.13 Modelo Entidad-Relación

El modelo entidad-relación representa los componentes de un proceso de negocio y sus relaciones. Tiene entidades, atributos y relaciones que se muestran gráficamente en un diagrama. (ESIC Business & Marketing School., s. f.)

#### 4.2.14 Arquitectura de Software

La arquitectura de software es la estructura del sistema que se crea en etapas tempranas del desarrollo para guiar el diseño y satisfacer los atributos de calidad como desempeño, seguridad y modificabilidad. (Arquitectura de Software, s. f.)

## 5. Requerimientos

### 5.1 Análisis de Requerimientos y Diseño de Funcionalidades en el Desarrollo de Tareas Asignadas

Para el desarrollo de las tareas asignadas, es fundamental llevar a cabo un exhaustivo análisis de los requerimientos funcionales y no funcionales del proyecto. Esto implica revisar detalladamente los casos de uso, identificar los actores involucrados y elaborar los correspondientes diagramas de secuencia y diagramas de actividades. Este proceso permite comprender a fondo el comportamiento esperado del sistema y la interacción entre sus diferentes componentes. Al analizar estos elementos, se establecen las bases sólidas para el diseño e implementación de las funcionalidades requeridas, garantizando que el desarrollo cumpla con los objetivos planteados y las expectativas de los usuarios.

#### 5.1.1 Home del Módulo Configuración General

- **Descripción:** Como usuario administrador de la plataforma de servicios UIS, requiero una pantalla de inicio para el módulo de configuración general, que sea común a los diferentes tipos de administradores y que todos los usuarios con permisos de administración, sin importar su tipo y menús habilitados, puedan converger allí al navegar el módulo de configuración General.

- **Actores:** Usuarios administradores.
- **Precondición:** El usuario debe tener acceso al módulo de configuración general para visualizar el home.

- **Postcondición:** Acceso al módulo y a todas sus funcionalidades.

### Requerimientos funcionales

1. La pantalla de inicio del módulo de configuración general debe mostrar de forma clara y organizada la información relevante para todos los usuarios que puedan acceder a dicho módulo, incluyendo un resumen de la funcionalidad de este menú principal.

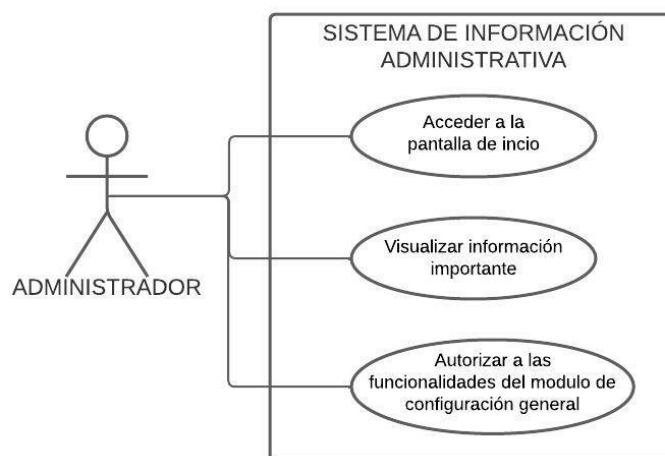
### Requerimientos no funcionales

- **Usabilidad:** La pantalla principal debe ser intuitiva y fácil de usar para todos los usuarios, incluyendo aquellos que no tienen experiencia en la plataforma de servicios UIS.
- **Adaptabilidad:** La pantalla principal debe ser adaptable a diferentes tipos de dispositivos y tamaños de pantalla, de modo que los usuarios puedan acceder a ella desde diferentes dispositivos y navegadores.
- **Mantenibilidad:** La pantalla principal debe ser fácil de mantener y actualizar, para asegurar que siempre esté actualizada con la información y las funcionalidades más recientes de la plataforma.

### Diagrama de caso de uso

**Figura 5.**

*Diagrama de casos de uso de la Home de configuración general.*

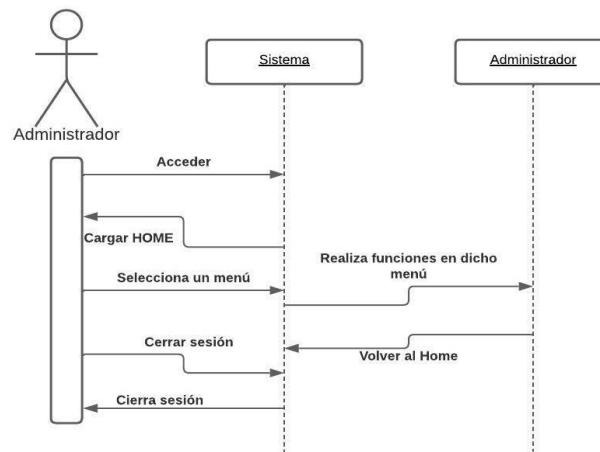


En este caso de uso, los administradores pueden acceder a la pantalla de inicio de la configuración general, donde pueden visualizar información importante y tienen autorización a las funcionalidades principales del módulo.

### Diagrama de secuencias

**Figura 6.**

*Diagrama de secuencias de la HOME de configuración general.*

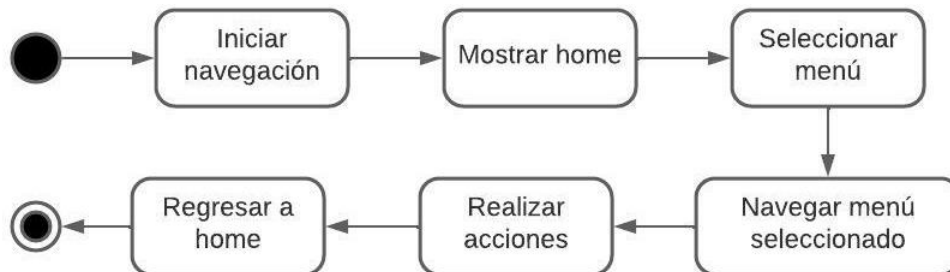


Este diagrama muestra el flujo de acciones que puede realizar un administrador en el módulo de configuración general, desde la selección de opciones en el home hasta la realización de acciones en otras pantallas y el regreso al home.

### Diagrama de actividades

**Figura 7.**

*Diagrama de actividades de la HOME de configuración general.*



El proceso comienza con la iniciación de la navegación, luego se muestra el *home* y el usuario selecciona el menú al que desea acceder. A continuación, se navega al menú seleccionado y se realizan las acciones correspondientes. Finalmente, se regresa al *home*.

### ***5.1.2 Acción de Generar Nueva Contraseña en el Listado de Usuarios***

- **Descripción:** Como administrador de la plataforma, necesito una opción para generar una nueva contraseña para un perfil, a fin de poder activar cuentas, solucionar problemas de acceso, y situaciones similares. Al utilizar esta opción, la nueva contraseña será enviada al correo electrónico registrado en el perfil. En caso de que el usuario no tenga un correo personal registrado, se utilizará el correo institucional. Si el cliente no cuenta con correo registrado, no se podrá realizar la acción.

- **Actores:** Usuarios y administradores.

- **Precondición:** Usuario con permisos de navegación para acceder al módulo de configuración general y usuario con correo electrónico registrado en el sistema, ya sea personal o institucional.

- **Postcondición:** Generación de una nueva contraseña para el usuario y envío de la misma al correo electrónico registrado en el sistema, si este cuenta con uno personal o institucional. Si el usuario no tiene un correo registrado, la opción de generar una nueva contraseña no estará disponible.

#### **Requerimientos funcionales**

1. El sistema debe proporcionar una opción clara y fácilmente accesible para que el administrador pueda generar una nueva contraseña para un perfil específico. Esta opción debe estar disponible en el listado y búsqueda de usuarios.

2. Al utilizar la opción de generar una nueva contraseña, el sistema debe validar si el perfil del usuario tiene un correo electrónico registrado. Si es así, la nueva contraseña debe ser enviada automáticamente a esa dirección de correo.

3. En caso de que el usuario no tenga un correo personal registrado, el sistema debe utilizar el correo institucional del usuario para enviar la nueva contraseña. Sin embargo, si el cliente no cuenta con ningún correo electrónico registrado, la acción de generar una nueva contraseña no debe ser posible y se deberá mostrar un mensaje de error indicando que el usuario debe registrar al menos un correo para poder llevar a cabo esta acción.

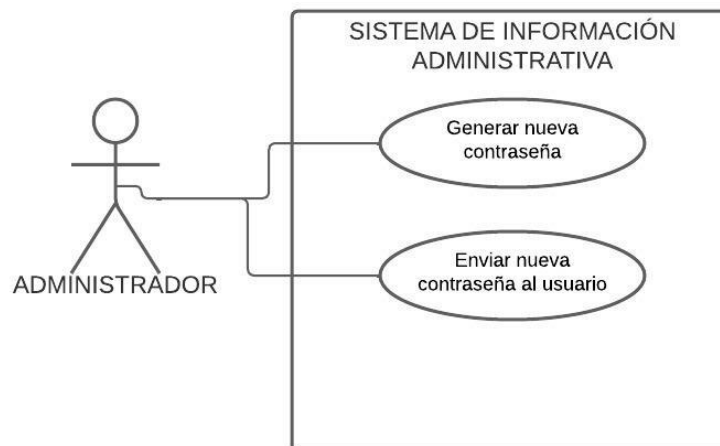
### Requerimientos NO funcionales

- **Usabilidad:** La interfaz debe ser intuitiva y fácil de usar para los administradores. El proceso de generación de la nueva contraseña debe ser claro y comprensible.
- **Mantenibilidad:** El código debe ser fácil de mantener y actualizar. Se deben seguir buenas prácticas de programación y documentar el proceso de generación de contraseñas para facilitar su actualización y corrección de errores en el futuro.

### Diagrama de casos de uso

#### Figura 8.

*Diagrama de casos de uso de la acción de generar nueva contraseña*

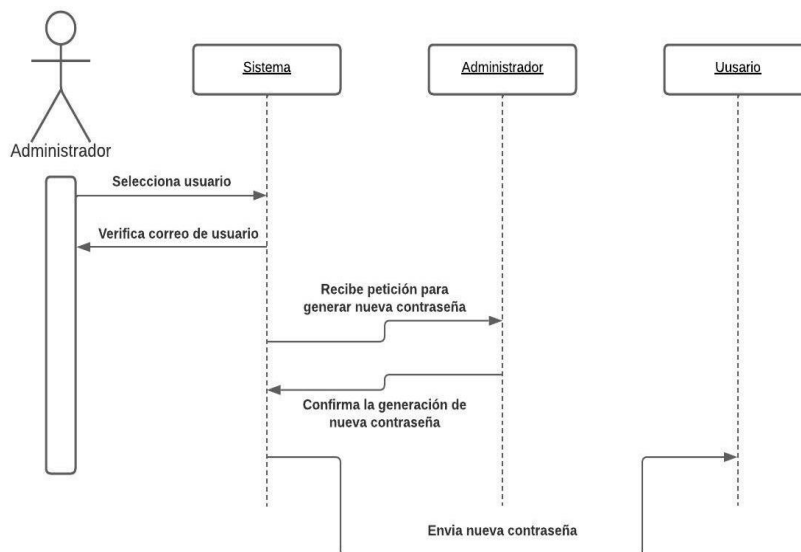


En este diagrama de caso de uso, podemos observar las acciones que hará el usuario para la nueva generación de contraseña. El actor es el Administrador, quien selecciona al usuario al cual se generará la nueva contraseña y confirma la generación de la misma. Por su parte, el sistema recibe la petición, verifica al cliente y su correo, genera la nueva contraseña y la envía al correo de dicho usuario. Estos pasos son necesarios para definir el alcance del desarrollo.

### Diagrama de secuencias

**Figura 9.**

*Diagrama de secuencias de acción de generar nueva contraseña.*

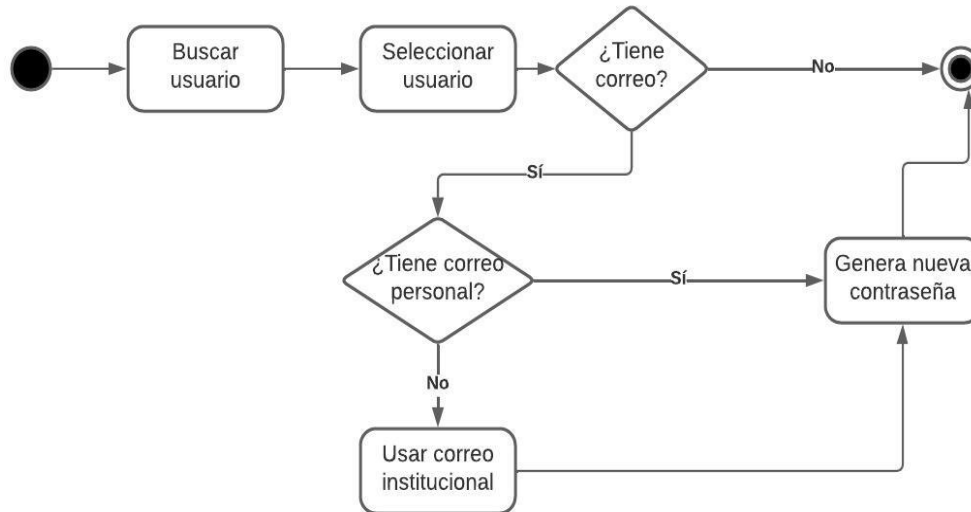


En este diagrama, el actor es el Administrador, quien selecciona al usuario al cual se generará la nueva contraseña y confirma la creación de la misma. Por su parte, el sistema recibe la petición para generar la nueva *password*, verifica al usuario, genera la nueva contraseña y la envía al correo del mismo.

### Diagrama de actividades

**Figura 10.**

*Diagrama de actividades de acción de generar nueva contraseña.*



En el diagrama de actividades, podemos ver el paso a paso que se necesita para generarle una nueva contraseña. Es vital para la producción de la reciente *password* tener un correo registrado, ya sea el personal o el institucional.

En caso de no tener un *email* asignado, no se puede proceder a realizar la acción de crear la próxima *password*.

#### 5.1.3 Pintar Menús de Acceso Directo en la Barra Superior Derecha

- **Descripción:** como usuario final del sistema, necesito tener acceso rápido a las funciones transversales sin salir del módulo en el que estoy trabajando. Por lo tanto, deseo poder crear menús de acceso directo a estas funciones y que se muestren como íconos en la barra superior derecha de la interfaz.

- La creación y edición de estos menús debe incluir una opción para marcarlos como menús directos, lo que permitirá que aparezcan en la barra superior. Además, debo contar con los permisos de navegación necesarios para crear y gestionar estos menús.

- **Actores:** Usuarios.
- **Precondición:** Usuario con permisos de navegación para manejar dichos menús.
- **Postcondición:** Menús de acceso directo creados y visibles en la barra superior derecha de la interfaz de usuario para facilitar el acceso a funciones transversales de la plataforma.

### **Requerimientos funcionales**

1. El sistema debe proporcionar una funcionalidad para crear menús de acceso directo a funciones transversales desde el módulo en el que el usuario está trabajando. Estos menús deben poder ser personalizados y configurados según las preferencias del usuario.

2. Al crear o editar un menú de acceso directo, el sistema debe permitir al usuario marcarlo como "menú directo" para que aparezca como un ícono en la barra superior derecha de la interfaz.

3. Para acceder a las funciones transversales y crear o gestionar los menús de acceso directo, el usuario final debe tener los permisos de navegación adecuados, garantizando que solo los usuarios autorizados puedan realizar estas acciones.

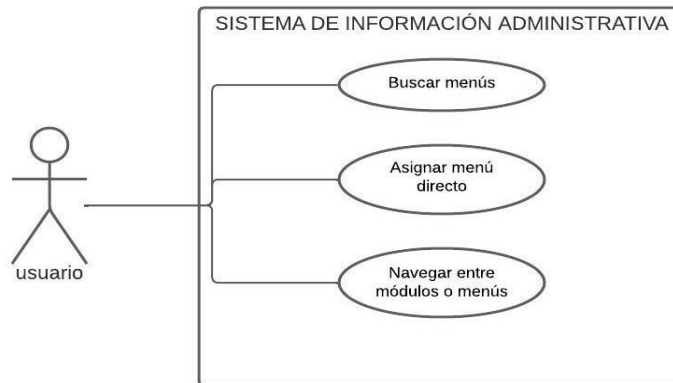
### **Requerimientos NO funcionales**

- **Usabilidad:** Los íconos de los menús deben ser fácilmente reconocibles y representativos de su función correspondiente para que los usuarios puedan encontrarlos intuitivamente.

## Diagramas de caso de uso

**Figura 11.**

*Diagrama de casos de uso de pintar menús de acceso directo.*

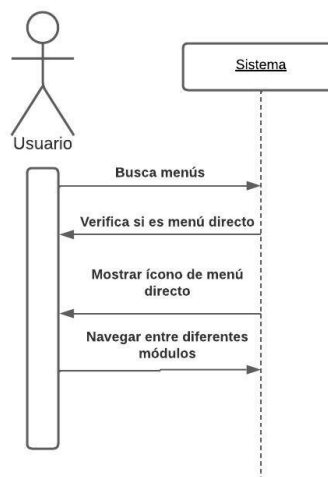


En el diagrama tomamos los posibles casos de uso para poder realizar la acción de “pintar” menús de acceso directo. Este desarrollo se guía principalmente en la gestión de menús.

## Diagramas de secuencia

**Figura 12.**

*Diagrama de actividades de pintar menús de acceso directo.*

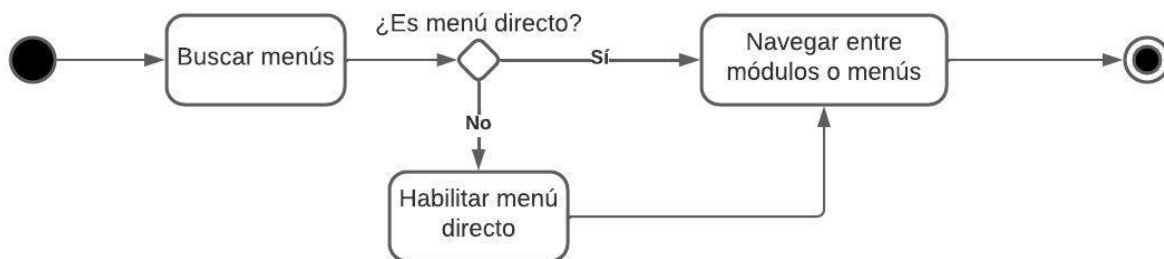


Este diagrama de secuencia ilustra cómo el usuario interactúa con los menús de acceso directo en la plataforma, desde la búsqueda de los menús hasta la navegación entre los diferentes módulos y menús.

### Diagrama de actividades

**Figura 13.**

*Diagrama de actividades de pintar menús de acceso directo.*



En este diagrama de actividades, la acción principal es buscar menús a los que el usuario tenga acceso, a partir de ahí, podemos preguntar si es menú directo o no, si es así, ya debe aparecer como acceso directo para poder navegar entre diferentes módulos o menús, en caso de no, se habilita la opción de *Rigth top menu* y, automáticamente, el cliente puede moverse entre diferentes rutas de navegación.

#### 5.1.4 Zonificar el Menú de Navegación Lateral Izquierdo

- **Descripción:** Como usuario del SIA, necesito que el menú lateral izquierdo muestre claramente los permisos de navegación asociados a mi rol, así como los permisos adicionales que se me hayan otorgado de forma individual. Además, necesito que el menú muestre claramente la diferencia entre los permisos otorgados por mi rol y los permisos otorgados directamente a mí como usuario.

- **Actores:** Usuarios del SIA.

- **Precondición:** Usuario registrado en el sistema y con un rol asignado que posee permisos de navegación para manejar el menú lateral izquierdo.
- **Postcondición:** Usuario que puede diferenciar claramente los permisos de navegación asociados a su rol y los permisos adicionales otorgados a sí mismo, lo que le permite entender sus privilegios de usuario y manejar los menús otorgados de manera efectiva.

### **Requerimientos funcionales**

1. El menú lateral izquierdo del SIA debe mostrar de manera clara y visible los permisos de navegación asociados al rol del usuario, resaltando las funcionalidades que están disponibles debido a su rol específico.
2. El menú debe mostrar de forma distintiva los permisos adicionales que se le han otorgado de manera individual al usuario, diferenciándolos claramente de los permisos inherentes a su rol.
3. Los permisos de navegación deben ser presentados de manera organizada y estructurada en el menú lateral, facilitando la comprensión y navegación para el usuario.
4. Si un usuario pierde ciertos permisos debido a cambios en su rol o a la revocación de permisos individuales, el menú debe reflejar estos cambios y ajustar las opciones disponibles en consecuencia.

### **Requerimientos no funcionales**

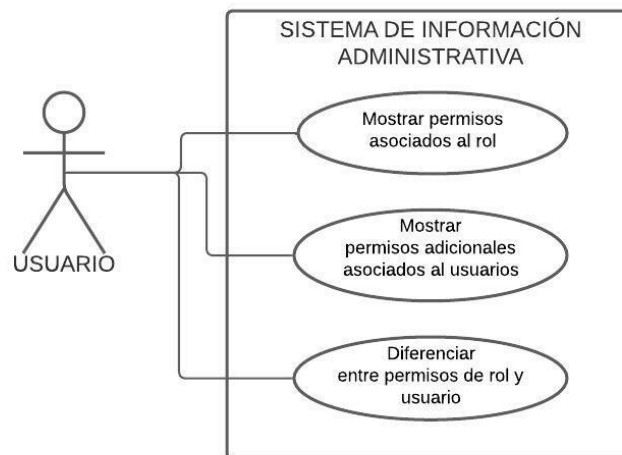
- **Usabilidad:** El diseño de la zonificación del menú lateral izquierdo debe ser intuitivo y fácil de entender para los usuarios, lo que permitirá una navegación clara y eficiente a través de los diferentes módulos y opciones disponibles.

- **Accesibilidad:** La zonificación del menú lateral izquierdo debe ser compatible con diferentes dispositivos y navegadores web, garantizando una experiencia de usuario uniforme independientemente del equipo utilizado para acceder al sistema.
- **Mantenibilidad:** Debe ser fácil de actualizar y mantener, para que los permisos de navegación otorgados a los usuarios puedan ser modificados y actualizados de manera rápida y sencilla.

### Diagrama de casos de uso

**Figura 14.**

*Diagrama de casos de uso sobre la zonificación del menú lateral izquierdo.*

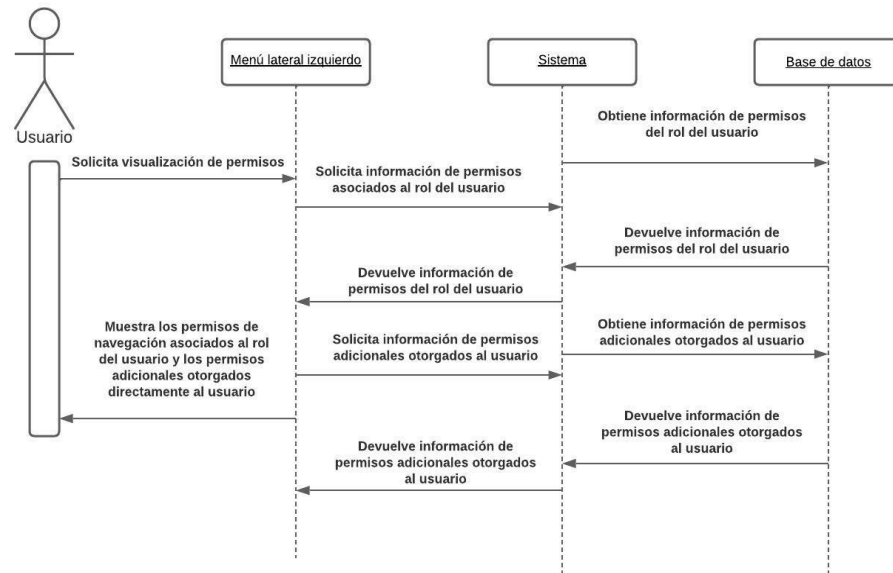


En este diagrama, el actor principal es el "Usuario". El actor puede visualizar los permisos de navegación asociados a su rol, así como los permisos adicionales otorgados de forma individual. El caso de uso permite diferenciar claramente entre los permisos otorgados por el rol y los permisos otorgados directamente al usuario. Después de visualizar los permisos, el usuario interactúa con el menú para realizar las acciones correspondientes.

## Diagrama de secuencias

**Figura 15.**

*Diagrama de secuencias sobre la zonificación del menú lateral izquierdo.*



En este diagrama, el usuario del SIA solicita la visualización de los permisos de navegación del menú lateral izquierdo.

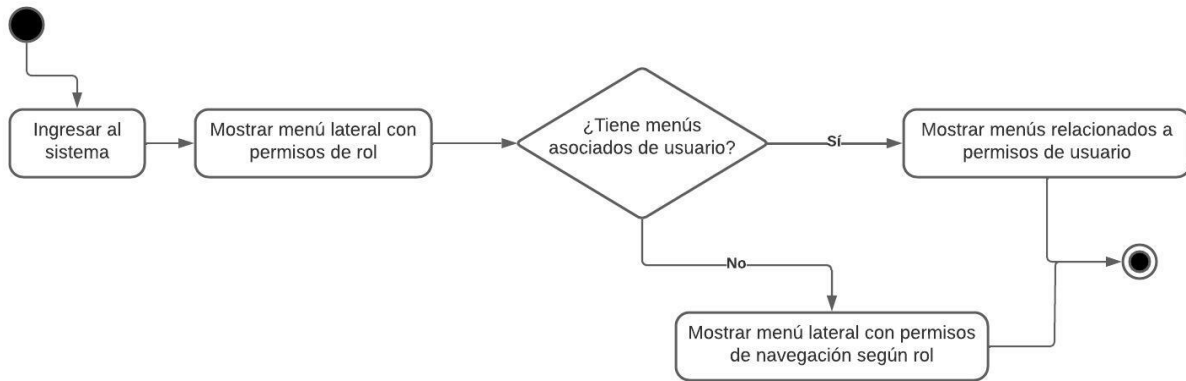
Luego, el menú lateral izquierdo hace dos solicitudes al sistema para obtener la información de permisos asociados al rol del usuario y los permisos adicionales otorgados directamente al usuario.

El sistema a su vez hace dos solicitudes a la base de datos para obtener la información de permisos, y finalmente el menú lateral izquierdo muestra los permisos al usuario del SIA.

### Diagrama de actividades

**Figura 16.**

*Diagrama de actividades sobre la zonificación del menú lateral izquierdo.*



En el diagrama de actividades, el usuario tiene un rol asignado con permisos de navegación para el menú lateral. Al entrar al sistema, se muestra el menú lateral. Luego, se verifica si el usuario tiene permisos adicionales otorgados de forma individual y si es así, se muestran estos permisos adicionales en el menú lateral junto con los permisos asociados a su rol. Finalmente, se llega al fin del proceso.

#### **5.1.5 Buscador en el Menú de Tercer Nivel**

- Descripción:** Como usuario de la nueva plataforma, necesito una función de filtrado en el menú vertical de acordeones para encontrar la opción que deseo de manera más rápida y fácil. El filtrado debe aplicarse directamente a los menús que se muestran al usuario y su rol actual a partir del tercer nivel, y debe ejecutarse desde el lado del cliente. Al escribir la primera letra en el filtro, se deben mostrar las hojas (opciones finales del árbol) que contengan la cadena de caracteres ingresada por el usuario, sin importar si son mayúsculas o minúsculas. Además, se debe mostrar la cadena de ancestros hasta el tercer nivel para facilitar la navegación. Cuando se acceda al menú

buscado, este debe permanecer resaltado en el color utilizado por el sistema para que el usuario pueda identificar fácilmente su ubicación en el menú.

- **Actores:** Usuarios.
- **Precondición:** El usuario debe tener permisos de navegación para acceder al menú.
- **Postcondición:** El usuario debe tener acceso por rol y usuario para navegar por las

rutas permitidas.

### **Requerimientos funcionales**

1. Implementar una función de filtrado en el menú vertical de acordeones que permita al usuario encontrar opciones de manera rápida y sencilla.
2. El filtrado debe aplicarse a partir del tercer nivel de menús, teniendo en cuenta el rol actual del usuario, para mostrar solo las opciones relevantes para él.
3. El filtrado debe realizarse en el lado del cliente, sin necesidad de recargar la página, para lograr una experiencia de usuario fluida y ágil.
4. Al escribir la primera letra en el filtro, se deben mostrar las hojas del árbol que contengan la cadena de caracteres ingresada, ignorando mayúsculas y minúsculas.
5. La visualización del menú filtrado debe incluir la cadena de ancestros hasta el tercer nivel para proporcionar contexto y facilitar la navegación.
6. Cuando el usuario acceda a un menú filtrado, este debe resaltarse con un color distintivo para que el usuario pueda identificar fácilmente su ubicación en el menú.

### **Requerimientos NO funcionales**

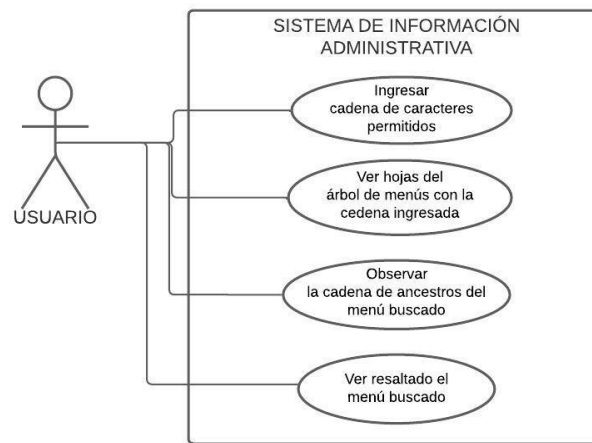
- **Usabilidad:** la función de filtrado debe ser fácil de usar e intuitiva para que los usuarios puedan encontrar rápidamente las opciones que desean.

- **Eficiencia:** el filtrado de menús debe ser rápido y no afectar el rendimiento general del sistema.
- **Compatibilidad:** el filtrado de menús debe funcionar en diferentes navegadores web y dispositivos.
- **Mantenibilidad:** el código del filtrado de menús debe ser fácil de mantener y actualizar para el equipo de desarrollo.

### Diagrama de casos de uso

**Figura 17.**

*Diagrama de casos de uso del buscador de menú de tercer nivel.*

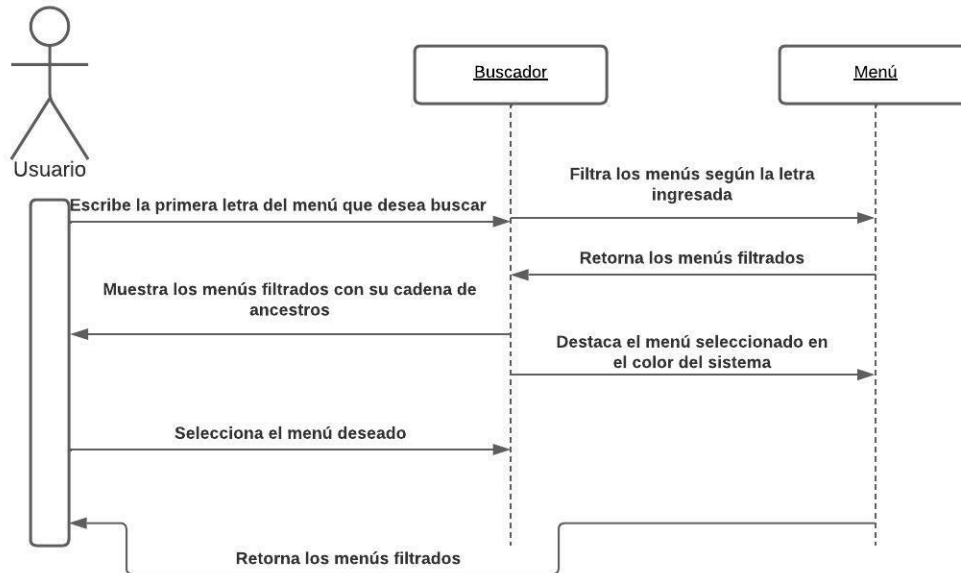


En este diagrama, el actor principal es el "Usuario". El actor puede ingresar una cadena de caracteres para filtrar los menús del menú vertical a partir del tercer nivel. Se mostrarán las hojas (opciones finales del árbol) que contengan la cadena ingresada, sin importar mayúsculas o minúsculas. Además, se mostrará la cadena de ancestros hasta el tercer nivel para facilitar la navegación. El menú buscado se resaltará en el color utilizado por el sistema. Después de filtrar el menú, el usuario interactúa con él para acceder a las rutas permitidas.

## Diagrama de secuencias

**Figura 18.**

*Diagrama de secuencias del buscador de menús.*

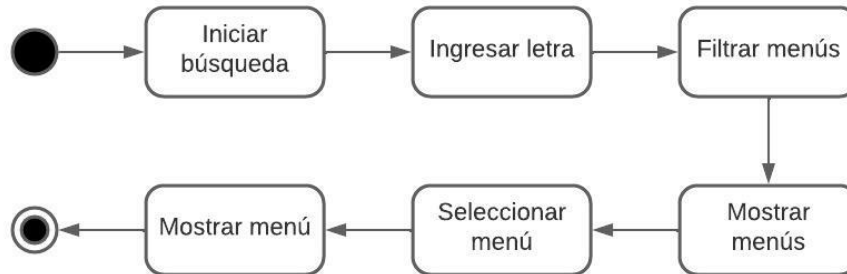


En este diagrama de secuencia se muestra la interacción entre el buscador de menú y la selección de una opción por parte del usuario. El buscador filtra automáticamente las opciones del menú vertical de acordeones a medida que el usuario escribe una letra. Luego, se muestran las opciones filtradas, junto con su cadena de ancestros hasta el tercer nivel para facilitar la navegación. Una vez que el usuario selecciona la opción deseada, el buscador la resalta en el color utilizado por el sistema para identificar fácilmente su ubicación en el menú. Finalmente, el menú muestra la opción seleccionada al usuario.

Diagrama de actividades

**Figura 19.**

*Diagrama de actividades del buscador de menús.*



En este diagrama, se puede observar el proceso que el sistema sigue para filtrar el menú vertical de acordeones. El proceso comienza cuando el usuario escribe la primera letra en el filtro, lo que activa la función de filtrado en el lado del cliente. Luego, se busca en el árbol de menú del usuario y se muestran las hojas que contienen la cadena de caracteres ingresada por el usuario.

Además, se muestra la cadena de ancestros hasta el tercer nivel para facilitar la navegación del usuario. Cuando se accede al menú buscado, este permanece resaltado en el color utilizado por el sistema para que el usuario pueda identificar fácilmente su ubicación en el menú.

Es importante mencionar que, para que el usuario pueda utilizar esta función de filtrado, debe tener permisos de navegación para acceder al menú. Además, una vez que se accede al menú buscado, el usuario debe tener acceso por rol y usuario para navegar por las rutas permitidas.

### ***5.1.6 Estabilización de Test de Frontend***

- Descripción: Se requiere estabilizar los módulos "Core" y "Authorization" para que pasen los pipes de test unitarios utilizando las herramientas Jasmine y Karma.
- Actores: Equipo de operaciones encargado del desarrollo de los módulos "Core" y "Authorization".

- Precondiciones: El equipo de operaciones debe contar con acceso y permisos necesarios para realizar las pruebas unitarias de los módulos y tener conocimientos en el uso de las herramientas Jasmine y Karma.

- Postcondiciones: Los módulos "Core" y "Authorization" deben pasar los pipes de test unitarios de manera satisfactoria, permitiendo la integración continua y un desarrollo más eficiente.

#### Requerimientos funcionales

1. Estabilizar el módulo "Core" para que pase los pipes de test unitarios utilizando las herramientas Jasmine y Karma.

2. Estabilizar el módulo "Authorization" para que pase los pipes de test unitarios utilizando las herramientas Jasmine y Karma.

3. Realizar pruebas unitarias exhaustivas en ambos módulos para garantizar la calidad y fiabilidad del código.

4. Implementar correcciones y ajustes en los módulos según los resultados de las pruebas unitarias para asegurar que cumplan con los estándares y requerimientos establecidos.

5. Documentar los procesos de estabilización y las soluciones aplicadas para facilitar el mantenimiento y futuras actualizaciones de los módulos.

#### Requerimientos NO funcionales

- Calidad del código: el código debe cumplir con los estándares de calidad establecidos por los principios de *clean code*.

- Estabilidad: los módulos de desarrollo deben estar estabilizados antes de permitir subir nuevos cambios.

- Test unitarios: se deben crear y ejecutar test unitarios para garantizar el correcto funcionamiento del código.
- Bloqueo de subidas: se deben bloquear las subidas de nuevos cambios que no cumplan con los estándares de calidad establecidos.

Para el desarrollo de esta HU, se usaron dos herramientas, Jasmine y Karma.

Jasmine es un marco de desarrollo basado en el comportamiento para probar código JavaScript. No depende de ningún otro *framework* JavaScript. No requiere un DOM. Y tiene una sintaxis limpia y obvia para que puedas escribir pruebas fácilmente. (Jasmine Documentation, s. f.)

Karma es un producto directo del equipo AngularJS de luchar para probar sus propias características del marco con las herramientas existentes. Como resultado de esto, crearon Karma y lo han trasladado a Angular como el ejecutor de pruebas predeterminado para las aplicaciones creadas con la CLI de Angular. (Morgan, 2020)

#### Organizar las pruebas

A continuación, se presenta información sobre cómo organizar y realizar pruebas en aplicaciones Angular utilizando la herramienta Jasmine y el ejecutor de pruebas Karma, integrados en Angular CLI.

Al crear un proyecto con la CLI de Angular, se descargan e instalan automáticamente todas las herramientas necesarias para realizar pruebas con Jasmine. Además, los proyectos creados con la CLI ya están configurados y listos para ejecutar pruebas. Para iniciar las pruebas, solo es necesario ejecutar el comando "ng test" en la terminal.

Este comando inicia la aplicación en modo monitor y arranca el ejecutor de pruebas Karma. La consola de salida mostrará información detallada sobre el proceso de prueba, incluyendo los

resultados de las pruebas unitarias y de integración, así como cualquier error o advertencia encontrada durante la ejecución.

### Figura 20.

*Salida de consola usando comando ng test.*

```
02 11 2022 09:08:28.605:INFO [karma-server]: Karma v6.4.1 server started at
http://localhost:9876/
02 11 2022 09:08:28.607:INFO [launcher]: Launching browsers Chrome with
concurrency unlimited
02 11 2022 09:08:28.620:INFO [launcher]: Starting browser Chrome
02 11 2022 09:08:31.312:INFO [Chrome]: Connected on socket -LaEYvd2R7MdcS0-AAAB
with id 31534482
Chrome: Executed 3 of 3 SUCCESS (0.193 secs / 0.172 secs)
TOTAL: 3 SUCCESS
```

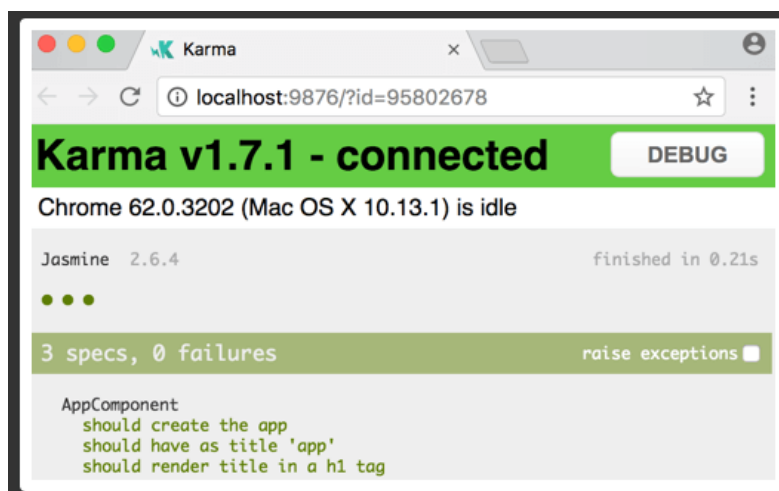
*Nota:* Angular. (SF). *Testing.* Tomado de: <https://angular.io/guide/testing>

El comando “ng test” crea la aplicación en modo monitor e inicia el ejecutor de pruebas Karma.

La consola se muestra así:

### Figura 21.

*Salida de consola en el navegador*



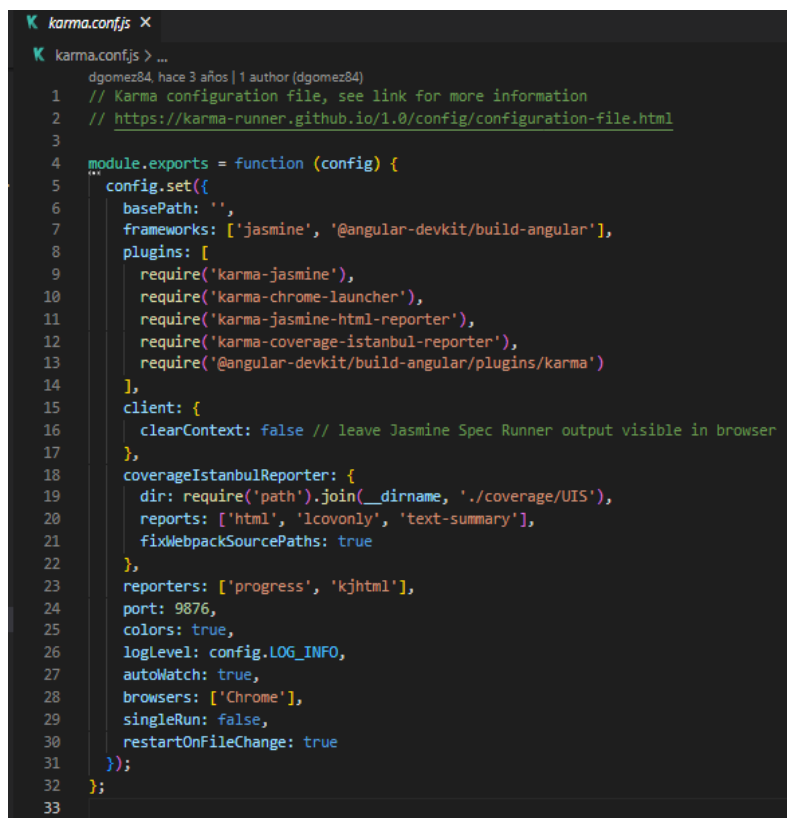
*Nota:* Angular. (SF). *Testing.* Tomado de: <https://angular.io/guide/testing>

## Configuración

La CLI de Angular facilita la configuración de Jasmine y Karma, ya que se encarga de crear toda la configuración en memoria en función de las opciones especificadas en el archivo `angular.json`. Además, si se desea personalizar la configuración de Karma, se puede generar un archivo `karma.conf.js` ejecutando el comando “`ng generate config karma`”. De esta manera, se puede ajustar la configuración de Karma para adaptarse a las necesidades específicas de la aplicación.

### Figura 22.

*Archivo `karma.conf.js` del proyecto de CORE.*



```
1 // Karma configuration file, see link for more information
2 // https://karma-runner.github.io/1.0/config/configuration-file.html
3
4 module.exports = function (config) {
5   config.set({
6     basePath: '',
7     frameworks: ['jasmine', '@angular-devkit/build-angular'],
8     plugins: [
9       require('karma-jasmine'),
10      require('karma-chrome-launcher'),
11      require('karma-jasmine-html-reporter'),
12      require('karma-coverage-istanbul-reporter'),
13      require('@angular-devkit/build-angular/plugins/karma')
14    ],
15    client: {
16      clearContext: false // leave Jasmine Spec Runner output visible in browser
17    },
18    coverageIstanbulReporter: {
19      dir: require('path').join(__dirname, './coverage/UIS'),
20      reports: ['html', 'lcovonly', 'text-summary'],
21      fixWebpackSourcePaths: true
22    },
23    reporters: ['progress', 'kjhtml'],
24    port: 9876,
25    colors: true,
26    logLevel: config.LOG_INFO,
27    autoWatch: true,
28    browsers: ['Chrome'],
29    singleRun: false,
30    restartOnFileChange: true
31  });
32 };
33
```

## Diagramas de caso de uso

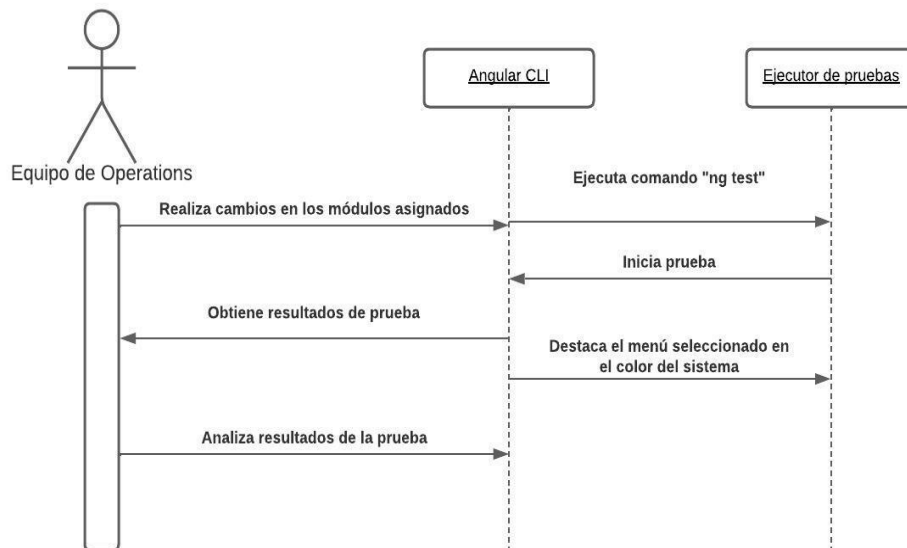
En este caso, no es necesario elaborar un diagrama de casos de uso, ya que la historia de usuario es bastante técnica y se enfoca en el proceso de prueba y organización del código. Los

casos de uso se utilizan principalmente para identificar y describir las interacciones entre los usuarios y el sistema, y en este caso, no hay una interacción directa con un usuario final. En lugar de ello, la historia se centra en el proceso de pruebas de una aplicación, por lo que se pueden utilizar otros tipos de diagramas, como diagramas de secuencia o diagramas de flujo, para representar el proceso técnico que se lleva a cabo.

### Diagramas de secuencia

#### Figura 23.

*Diagrama de secuencias de estabilización de test de frontend.*

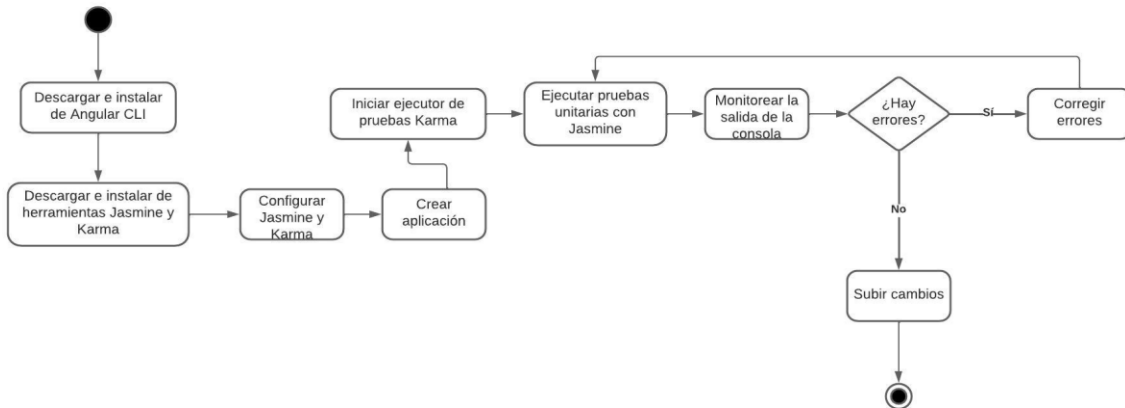


El diagrama de secuencias representa el proceso a seguir por cada integrante del equipo de *Operations* para la correcta estabilización de pruebas en el *frontend* de cada módulo asignado.

## Diagrama de actividades

**Figura 24.**

*Diagrama de actividades de estabilización de test de frontend.*



El diagrama de actividades representa el proceso de estabilización de módulos en el que trabaja el equipo de *operations* para que pasen los *pipes* de test unitarios. Inicia con la ejecución del comando "ng test", seguido por la creación de la aplicación en modo monitor y el inicio del ejecutor de pruebas Karma. Luego se ejecutan las pruebas y se verifica si pasan o fallan. Si las pruebas fallan, se procede a corregir los errores y se repite el proceso hasta que todas las pruebas pasen. Finalmente, se envía una notificación indicando que las pruebas han sido estabilizadas correctamente.

### 5.1.7 CRUD de Autoregistros

- Descripción: Como administrador de la nueva plataforma de la Universidad Industrial de Santander deseo poder generar un proceso básico de registro en esta plataforma que pueda ser usado como base para distintos procesos.

- Actores: Usuarios (Administradores).
- Precondición: Usuarios con permisos de navegación para acceder a la gestión de autoregistros.

- Poscondición: Generar links de autoregistros para facilitar el ingreso y registro al Sistema de Información Administrativa (SIA).

#### Requerimientos funcionales

1. Se creará una nueva vista para gestionar los distintos tipos de autoregistros en Configuración > General > Usuarios y accesos > Gestión de autoregistros.
2. Al ingresar a la vista de debe mostrar una tabla que contenga Nombre, descripción, fecha desde, fecha hasta y acciones de los autoregistros.
3. Los registros en los que la fecha hasta sea menor que la fecha actual deberá aparecer desactivados (las acciones siguen habilitadas, pero en gris).
4. La tabla debe tener la opción de ordenar por cualquiera de las columnas.
5. El ordenamiento por defecto será por fecha hasta (de menor a mayor).
6. Al dar click en crear autoregistro debe aparecer un modal con los campos para nombre, descripción, fecha desde, fecha hasta y un check para esquema registro.
7. Los campos obligatorios serán: Nombre, Descripción y fecha desde.
8. El nombre debe escribirse la primera en mayúscula y el resto en minúscula.
9. La descripción tiene un máximo de 400 caracteres.
10. La fecha desde se llenará por defecto con el día actual.
11. No se puede escoger una fecha desde anterior al día actual, ni posterior al día de la fecha hasta cuando sea escogida.
12. La fecha hasta aparece vacía por defecto.
13. No se puede escoger una fecha hasta anterior a la fecha desde.

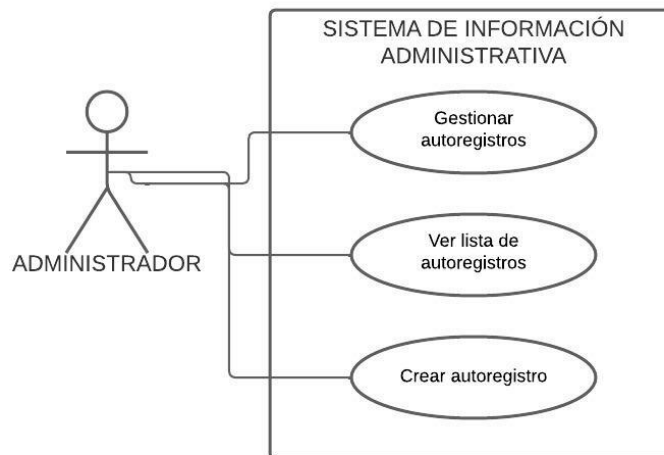
Requerimientos no funcionales:

- Usabilidad: La interfaz de usuario debe ser intuitiva y fácil de usar, de modo que los administradores puedan gestionar los autoregistros sin dificultad.
- Seguridad: El acceso a la gestión de autoregistros debe estar restringido a usuarios con permisos de administrador para garantizar la integridad de los registros.
- Mantenibilidad: El código del CRUD de autoregistros debe estar bien estructurado y documentado para facilitar futuras actualizaciones y mejoras.

Diagramas de caso de uso

### Figura 25.

*Diagrama de casos de uso de CRUD de autoregistro.*

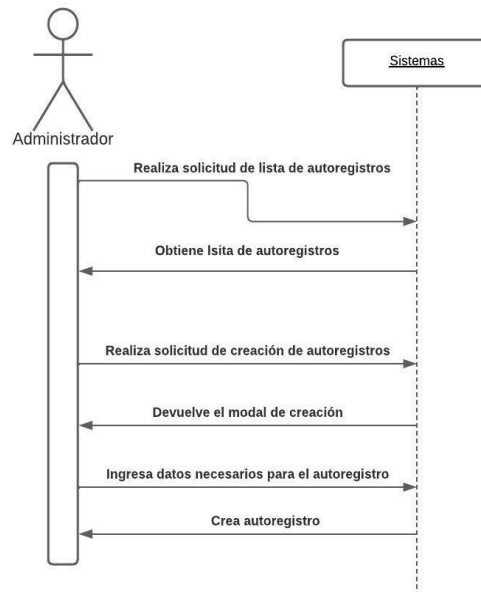


En este diagrama, el único actor es el Administrador, quien tiene la capacidad de gestionar los autoregistros. El caso de uso principal es "Gestionar Autoregistros", que incluye subcasos como "Ver lista de autoregistros" y "Crear Autoregistro". El Administrador puede ver la lista de autoregistros existentes y crear nuevos autoregistros.

## Diagramas de secuencia

**Figura 26.**

*Diagrama de secuencia de CRUD de autoregistro.*

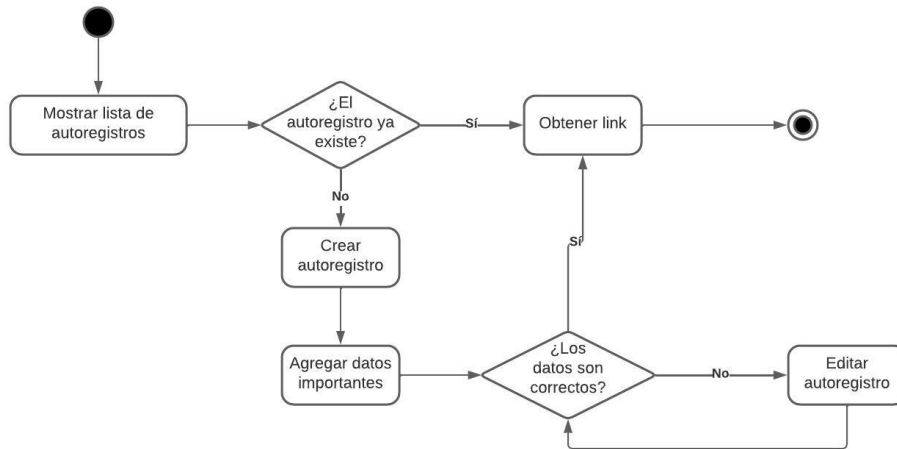


En este diagrama de secuencia, el Administrador interactúa con la interfaz de usuario para gestionar los autoregistros. Realiza una solicitud de lista de autoregistros, lo cual devuelve la lista de autoregistros existentes. Luego, realiza una solicitud de creación de autoregistro, lo cual muestra el modal de creación en la interfaz de usuario. El Administrador ingresa los datos del autoregistro y realiza una solicitud de creación con los datos ingresados. Finalmente, se devuelve la confirmación de que el autoregistro ha sido creado.

## Diagrama de actividades

**Figura 27.**

*Diagrama de actividades de CRUD de autoregistro.*



En este diagrama de actividades, el Administrador inicia el proceso de gestión de autoregistros. Se muestra la lista de autoregistros disponibles y el Administrador puede seleccionar uno para editarlo. Luego, se validan los campos obligatorios y las fechas del autoregistro. Si todo es válido, se crea el autoregistro y se muestra un mensaje confirmando la creación. Después, se actualiza la lista de autoregistros para reflejar los cambios realizados. Por último, se obtiene el link para compartir con el usuario.

## 6. Diseño

### 6.1 Arquitectura de la Renovación de los Sistemas de Información (RSI)

**Figura 28.**

*Tecnologías utilizadas en RSI.*



La arquitectura de microservicios es un enfoque de diseño de software que se ha vuelto cada vez más popular en los últimos años. En lugar de desarrollar una aplicación monolítica, los microservicios dividen el sistema en una colección de pequeños servicios independientes, cada uno responsable de una función específica del negocio. Estos servicios se ejecutan de manera autónoma y se comunican entre sí a través de API bien definidas.

Una característica destacada de la arquitectura de microservicios de RSI es la separación de los repositorios para el front-end y el back-end de cada módulo del sistema. Esto permite que los equipos de desarrollo trabajen de manera más eficiente y flexible, ya que pueden implementar

cambios en el front-end o en el back-end de forma independiente sin afectar el funcionamiento general del sistema.

Además, en la base de datos, se emplean diferentes esquemas para gestionar la información de manera más efectiva. Cada módulo del sistema tiene su propio esquema en la base de datos, lo que permite una mejor organización y mantenimiento de los datos asociados a ese módulo. Esta práctica también ayuda a evitar conflictos y problemas de rendimiento al tener múltiples equipos trabajando en diferentes partes del sistema.

GitLab desempeña un papel fundamental en este enfoque de desarrollo. Con su controlador de versiones y la posibilidad de tener varios repositorios para cada módulo y cada componente, GitLab facilita la gestión y el seguimiento de los cambios en el código. Las distintas ramas, como PREDEV, DEVELOP, PREPROD y PRODUCCIÓN, proporcionan un marco de trabajo para el desarrollo, las pruebas y la implementación de nuevas funcionalidades en el sistema.

## **6.2 Tecnologías Implementadas**

El conjunto de tecnologías con las que fueron desarrollados los componentes del proyecto es:

### **6.2.1 Angular**

Angular es un framework JavaScript potente, muy adecuado para el desarrollo de aplicaciones frontend modernas, de complejidad media o elevada. El tipo de aplicación JavaScript que se desarrolla con Angular es del estilo SPA (Single Page Application) o también las denominadas PWA (Progressive Web App). El framework Angular ofrece una base para el desarrollo de aplicaciones robustas, escalables y optimizadas, que promueve además las mejores prácticas y un estilo de codificación homogéneo y de gran modularidad (Angular, s. f.).

### **6.2.2 JAVA SPRINGBOOT**

Java Spring Framework (Spring Framework) es un marco de trabajo popular, de código abierto y de nivel empresarial para crear aplicaciones independientes y de nivel de producción que se ejecutan en la máquina virtual Java (JVM) (Education, I. C. ,2021).

Java Spring Boot (Spring Boot) es una herramienta que agiliza y facilita el desarrollo de aplicaciones web y microservicios con Spring Framework a través de tres capacidades básicas: la autoconfiguración, un enfoque orientado a la configuración y la capacidad de crear aplicaciones independientes. (Education, I. C., 2021).

### **6.2.3 ORACLE DB**

Oracle Database (Oracle DB) es un sistema de gestión de bases de datos relacionales (RDBMS) desarrollado originalmente en 1977 por Lawrence Ellison y otros desarrolladores. Es ampliamente reconocido como uno de los motores de bases de datos más confiables y utilizados para almacenar, organizar y recuperar datos manteniendo relaciones entre distintos tipos de datos. Los usuarios pueden acceder a los objetos de datos a través de un lenguaje de consulta estructurado (SQL) en un marco de base de datos relacional.

Oracle es una arquitectura escalable utilizada por empresas globales que gestionan y procesan datos a través de redes de área local y amplia, con su propio componente de red para permitir la comunicación. (Techopedia, 2021)

### **6.2.4 GITLAB**

GitLab es una plataforma de desarrollo de software colaborativo y un repositorio de código abierto. Es ampliamente utilizado para proyectos DevOps y DevSecOps, ofreciendo almacenamiento de código en línea, seguimiento de incidencias y capacidades de CI/CD. Es gratuito para particulares y proporciona un entorno para alojar diversas cadenas de desarrollo y

versiones. Además, permite a los usuarios inspeccionar y revertir el código anterior en caso de problemas. (Contributor, T., 2020)

### **6.2.5 TAIGA (*Gestión de Proyecto*)**

Taiga es una herramienta de gestión y colaboración en proyectos ágiles, diseñada para utilizar metodologías Scrum y Kanban, así como para gestionar incidencias. Es de código abierto y ofrece una plataforma flexible y adaptada para equipos que buscan una gestión eficiente y colaborativa en proyectos ágiles. (Toro, L., 2015)

## **6.3 Proceso de Desarrollo con GITLAB, KONG Y CI/CD**

El proyecto utiliza GitLab como controlador de versiones y repositorios, lo que permite gestionar de manera eficiente el control de cambios y la colaboración entre los miembros del equipo. Cada módulo del proyecto se divide en dos repositorios independientes: uno para el front-end y otro para el back-end. Esta estructura facilita la separación de responsabilidades y el desarrollo paralelo de ambos componentes.

Cada repositorio cuenta con un archivo ".gitlab-ci.yml", el cual desempeña un papel fundamental en la implementación de la integración y entrega continuas (CI/CD). Este archivo de configuración define los pasos y las acciones a ejecutar durante los pipelines. Los pipelines son flujos de trabajo automatizados que se activan en respuesta a eventos, como la confirmación de cambios en el repositorio.

Estos pipelines incluyen tareas como la compilación del código fuente, las pruebas automatizadas, la generación de artefactos y el despliegue a entornos de desarrollo, preproducción y producción. Gracias a los pipelines, se logra una mayor agilidad en el ciclo de desarrollo y se garantiza la calidad del software entregado.

Además, en el proceso de despliegue de las aplicaciones, se utiliza Kong como una plataforma de gestión de API. Kong se encarga de cuadrar las rutas de las API y proporcionar un control centralizado sobre ellas.

Permite definir políticas de acceso, autenticación, autorización y transformación de datos, lo que brinda flexibilidad y seguridad en el manejo de las solicitudes de las aplicaciones. Kong actúa como un *gateway* o intermediario entre las aplicaciones y los servicios, facilitando la comunicación y el enrutamiento eficiente de las peticiones.

En conclusión, la combinación de GitLab, los repositorios separados, los archivos de configuración CI/CD y el uso de Kong como gestor de API, son elementos clave en la arquitectura y el proceso de desarrollo del proyecto. Estas herramientas y prácticas promueven la colaboración, la automatización y la estandarización, permitiendo un desarrollo más eficiente, controlado y seguro de las aplicaciones del proyecto.

## **6.4 Modelos Entidad Relación (ER)**

### ***6.4.1 Modelo ER HOME de Configuración General***

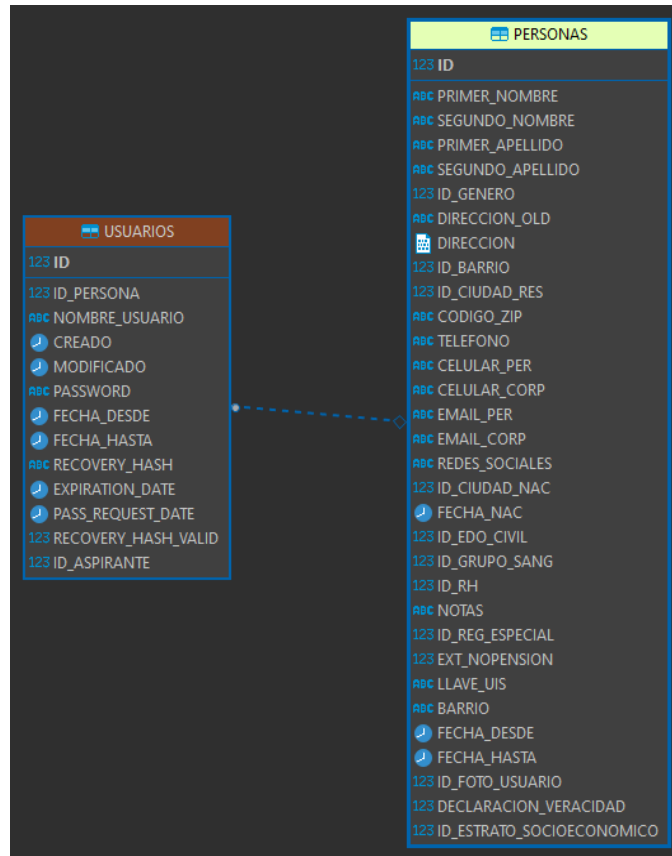
Desafortunadamente, el modelo entidad-relación se entiende como un diagrama que representa las entidades relevantes del sistema y sus relaciones, pero en este caso el Home del módulo de configuración general no está relacionado con las unidades del sistema, sino con la interfaz de usuario del sistema y navegación en la plataforma.

Por lo tanto, el modelo ER no se puede construir en la página de inicio del módulo de configuración general.

### 6.4.2 Modelo ER de Acción de Generar Nueva Contraseña

**Figura 29.**

*Diagrama Entidad Relación de la tabla Usuarios y Personas.*

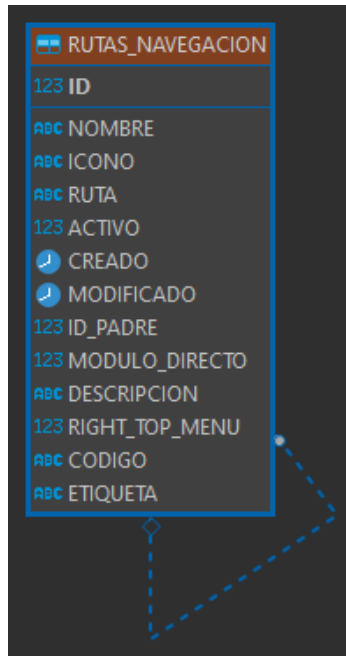


En el diagrama ER se puede observar que la tabla PERSONAS tiene una relación de uno a muchos con la entidad USUARIOS. Esto se debe a que una persona puede tener múltiples usuarios. Gracias a esta conexión, se pueden obtener tanto el correo personal como el correo institucional de un usuario, lo que es necesario para generar una nueva contraseña.

### 6.4.3 Modelo ER de Pintar Menús de Acceso Directo en la Barra Superior Derecha

**Figura 30.**

*Diagrama entidad-relación de las rutas navegación.*



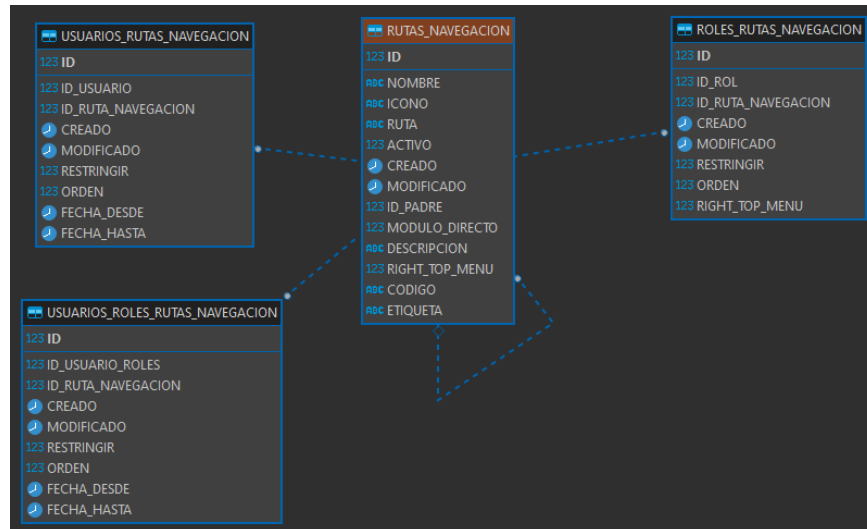
En el siguiente diagrama entidad-relación, se muestra la tabla de RUTAS\_NAVEGACIÓN con todas sus características. En este caso, nos interesa el atributo denominado RIGHT\_TOP\_MENU, el cual es un campo booleano que puede tener dos valores: 0 o 1. Cuando el valor es 0, indica que es falso, lo que significa que ese menú no será mostrado como menú directo.

Por otro lado, si el valor es 1, será verdadero, lo que indica que sí será un menú directo y aparecerá en la barra superior derecha.

#### 6.4.4 Modelo ER de Zonificar el Menú de Navegación Lateral Izquierdo

**Figura 31.**

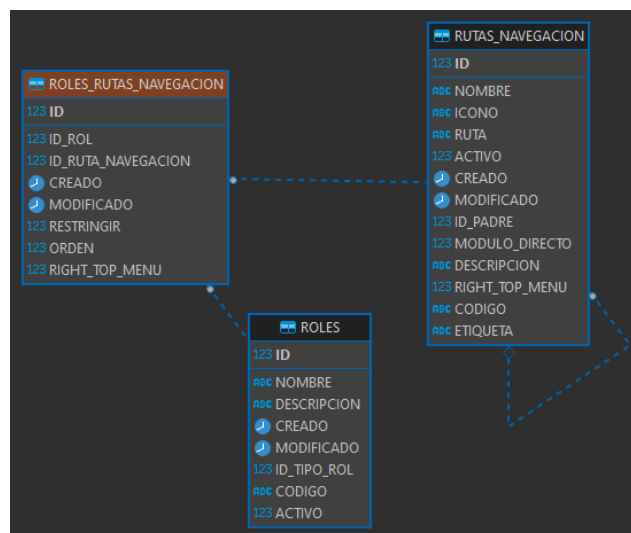
*Diagrama entidad-relación de las rutas navegación.*



Como podemos observar, el esquema RUTAS\_NAVEGACION se relaciona con las tablas ROLES\_RUTAS\_NAVEGACION, USUARIOS\_RUTAS\_NAVEGACION y USUARIOS\_ROLES\_RUTAS\_NAVEGACION.

**Figura 32.**

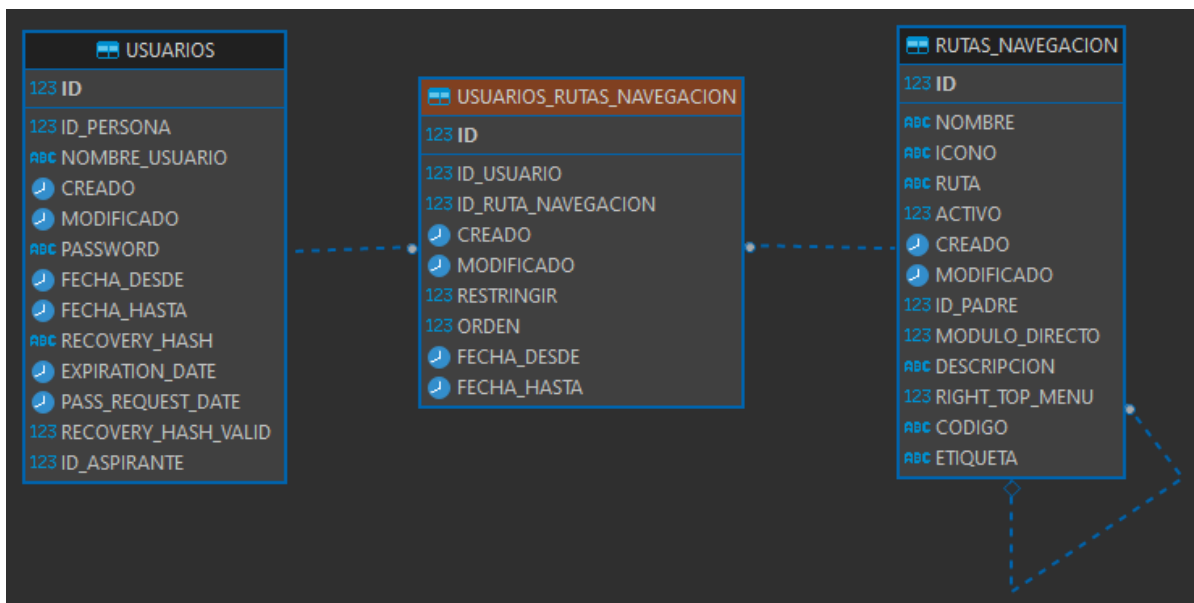
*Diagrama entidad-relación de los roles rutas navegación.*



La tabla ROLES\_RUTAS\_NAVEGACION tiene como finalidad establecer la conexión entre la entidad RUTAS\_NAVEGACION, encargada de manejar los menús o rutas del sistema, con el esquema ROLES, el cual registra los roles utilizados en el sistema. Esta tabla se ha diseñado para establecer una relación de muchos a muchos, ya que un rol puede tener permiso para acceder a múltiples rutas de navegación, y una ruta de navegación puede ser accedida por múltiples roles.

### Figura 33.

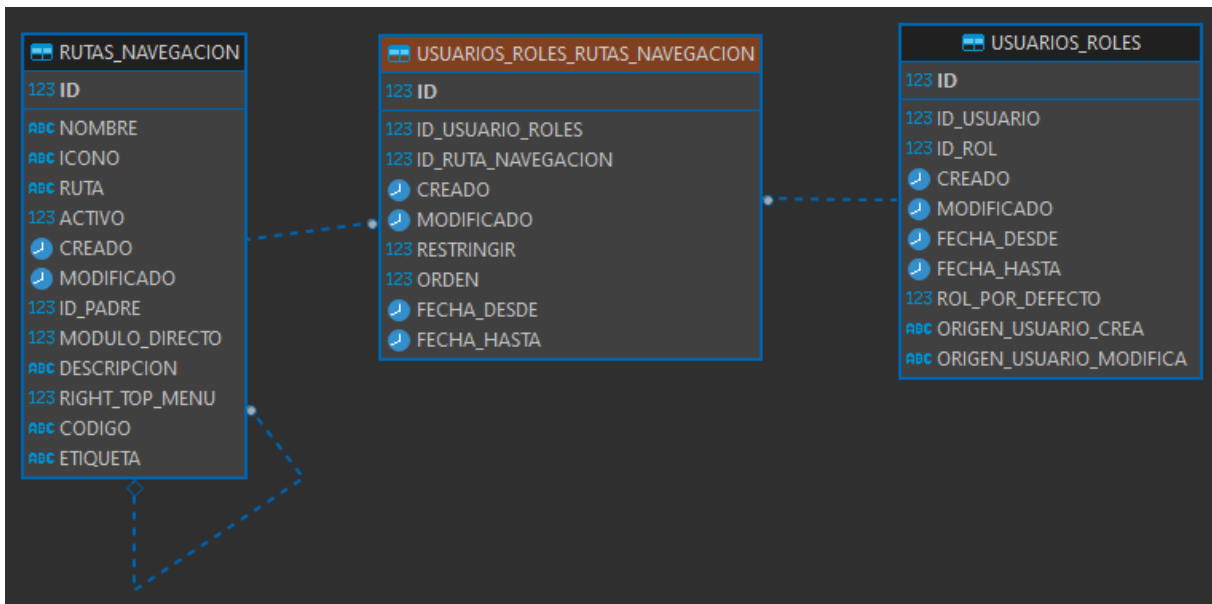
*Diagrama entidad-relación de los usuarios rutas navegación.*



La entidad USUARIOS\_RUTAS\_NAVEGACION tiene como objetivo hacer la conexión entre la tabla RUTAS\_NAVEGACION, la cual gestiona las rutas o menús del sistema, y el esquema USUARIOS, que registra a los usuarios del sistema. Esta entidad se ha diseñado para permitir que un usuario tenga permiso para acceder a una ruta de navegación, así como para que una ruta de navegación pueda ser accedida por un usuario en particular. Al igual que con la tabla ROLES\_RUTAS\_NAVEGACION, esta tabla también establece una relación muchos a muchos entre las entidades involucradas.

**Figura 34.**

*Diagrama entidad-relación de los usuarios roles rutas navegación.*



La tabla `USUARIOS_ROLES_RUTAS_NAVEGACION` se encarga de la conexión entre la entidad `RUTAS_NAVEGACION` y el esquema `USUARIOS_ROLES`, el cual registra tanto a los usuarios como los roles que estos poseen. En otras palabras, esta tabla actúa como una tabla intermedia y permite establecer una relación muchos a muchos entre usuarios y roles. De esta manera, un usuario puede tener varios roles y, a su vez, un mismo rol puede ser compartido por varios usuarios. Este tipo de relación es utilizado ampliamente en el proyecto.

#### **6.4.5 Modelo ER de Buscador de Menú de Tercer Nivel**

El buscador de menú de tercer nivel no requiere de un diagrama entidad relación porque no implica el manejo de una base de datos o almacenamiento de información. Se trata de una función de filtrado que se ejecuta del lado del cliente y que busca en los menús del sistema en tiempo real para mostrar opciones que coincidan con la cadena de caracteres ingresada por el usuario. No hay entidades ni relaciones que deban ser modeladas, ya que el proceso no involucra el manejo de datos estructurados o la creación de tablas de base de datos. Por lo tanto, el enfoque

es en la implementación de la lógica del filtrado y la interacción con la interfaz de usuario para mostrar los resultados de manera clara y efectiva.

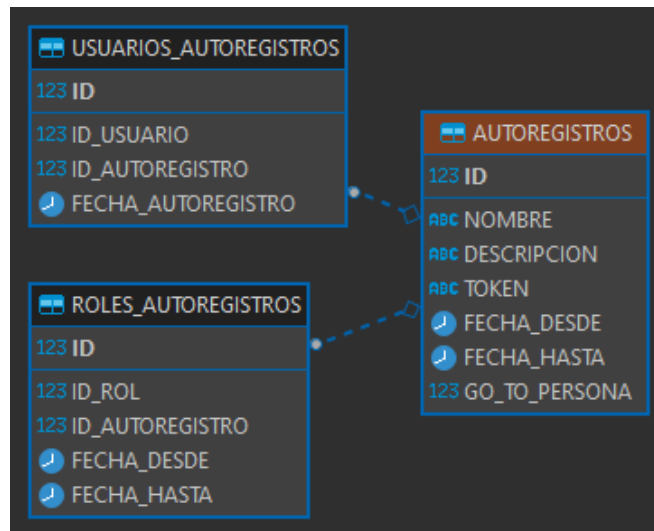
#### 6.4.6 Modelo ER de Estabilización de Test de Frontend

No es posible hacer un modelo entidad-relación para esta historia de usuario, ya que se trata de una tarea de pruebas de software, que se enfoca en la funcionalidad de los módulos Core y Authorization del sistema, los cuales no tienen relación directa con entidades o relaciones de datos en una base de datos. Por lo tanto, no se puede modelar una estructura de entidad y relación para estas tareas de pruebas, sino que se requiere un enfoque diferente para planificar y ejecutar las pruebas, como se describe en los diagramas de secuencia y actividades previamente diseñados.

#### 6.4.7 Modelo ER de Crud de Autoregistros

##### Figura 35.

Diagrama Entidad-Relación de autoregistro.



La tabla AUTOREGISTROS se relaciona con las entidades USUARIOS\_AUTOREGISTROS y ROLES\_AUTOREGISTROS. Estas entidades actúan como tablas intermedias para las clases USUARIOS y ROLES, respectivamente.

**Figura 36.**

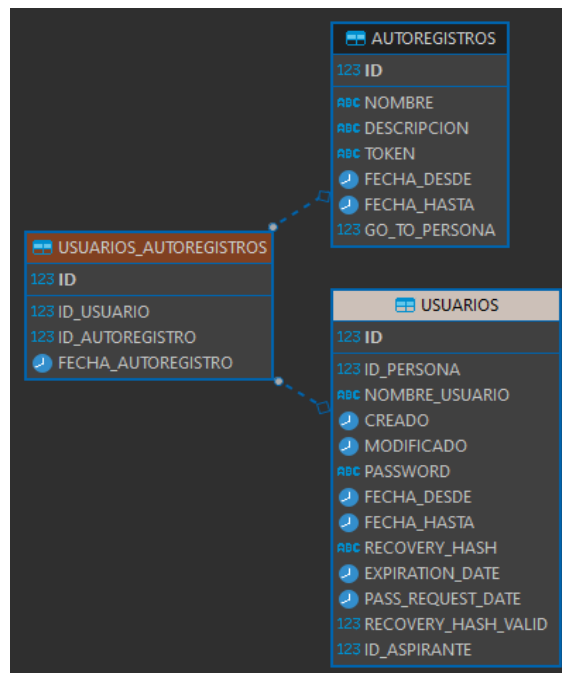
*Diagrama Entidad-Relación de autoregistro y Roles.*



Como se explicó anteriormente, la tabla ROLES\_AUTOREGISTROS actúa como una tabla intermedia entre AUTOREGISTROS y ROLES. Debido a la relación muchos a muchos, un rol puede estar asociado con varios autoregistros y viceversa.

**Figura 37.**

*Diagrama Entidad-Relación de autoregistro y usuarios.*



Para concluir, la tabla `USUARIOS_AUTOREGISTROS` se utiliza como una tabla intermedia entre `AUTOREGISTROS` y `USUARIOS`. Debido a la relación muchos a muchos, un usuario puede estar asociado con varios autoregistros, y a su vez, un autoregistro puede estar asociado a varios usuarios.

## 7. Implementación

### 7.1 Herramientas y Procedimientos para el Desarrollo del Proyecto

Como se ha mencionado previamente en el apartado de arquitectura, el proyecto se basa en el uso de diversas herramientas de desarrollo y mantenimiento de software, incluyendo:

- Angular, un *framework* para el *frontend*.
- Spring Boot, un *framework* para el *backend*.
- Oracle DB, la base de datos utilizada.
- GitLab, el controlador de versiones empleado.
- Taiga, el administrador de proyectos utilizado.

El primer paso para iniciar el desarrollo es obtener acceso al Taiga, el cual es otorgado por el Scrum Master del proyecto a través de un correo asignado para este propósito. Una vez que se obtienen los permisos en Taiga, el líder de cada equipo realiza su respectivo *planning* con su grupo de desarrolladores, estimando las historias de usuario (HU) que se incluirán en el *sprint*. A continuación, se definen las tareas asignadas a cada miembro del equipo. Por lo general, una historia de usuario se asigna a un desarrollador de *frontend* y otro de *backend*, o a veces a una sola persona que maneje ambos campos.

Una vez asignada la historia, se solicita al DevOps del equipo la creación de un usuario en GitLab. Este usuario es de vital importancia, ya que permite clonar los archivos del proyecto, aceptar *merge requests* (MR) y subir cambios a las ramas correspondientes en los distintos entornos (predev, develop, preprod y production). Una vez se cuenta con el acceso asignado, el siguiente paso es generar un token de acceso, el cual se utiliza para clonar los repositorios y también para instalar las bibliotecas o dependencias necesarias en los proyectos.

Además del usuario de GitLab, se requiere acceso a la base de datos, el cual es otorgado por el ingeniero a cargo. Este acceso proporciona permisos en las bases de datos de los entornos *predev* y *develop*, que son utilizados para realizar pruebas por parte de los desarrolladores y los especialistas en control de calidad (QA).

Una vez que los usuarios están listos, el desarrollador debe clonar el repositorio en el que va a trabajar. Por lo general, un equipo maneja dos repositorios, uno para el *frontend* y otro para el *backend*, aunque también existen grupos transversales que gestionan más de dos proyectos simultáneamente. El IDE más utilizado para la programación en el *backend* es IntelliJ, mientras que para el *frontend* se utiliza VSCode.

Para los desarrolladores de *backend*, hay dos archivos importantes: el primero es el `pom.xml`, que contiene las bibliotecas que se van a utilizar, y el segundo es el `application.properties`, donde se configura el entorno de trabajo para programar y realizar pruebas. Por otro lado, los desarrolladores de *frontend* utilizan el token de acceso generado anteriormente para ejecutar los comandos necesarios y así instalar las dependencias requeridas para un correcto funcionamiento en su máquina personal.

Antes de comenzar el desarrollo, se recomienda a los desarrolladores actualizar los cambios de la rama *predev*, ya que esta contiene las últimas modificaciones y es el entorno diseñado para los programadores. También se aconseja crear una rama con el número de la historia de usuario correspondiente para identificar más rápidamente los cambios agregados a la rama principal.

**Figura 38.***Tecnologías utilizadas en RSI.*

## 7.2 Subir Código al Repositorio

En este proyecto, se utilizó GitKraken Client como herramienta para subir el código al repositorio. GitKraken es una herramienta multiplataforma que simplifica el manejo de Git, mejorando la productividad. Integra GitHub, Bitbucket y GitLab, permitiendo arrastrar y soltar funcionalidades como merge, rebase y push de forma intuitiva. (Alex, 2021)

La subida de código al repositorio sigue los siguientes pasos:

1. Actualizar la rama principal de desarrollo, llamada "predev", para obtener los últimos cambios realizados por los programadores. Esto evita errores de código y asegura la integración adecuada de cambios nuevos y antiguos.

2. Crear una nueva rama local con el nombre "FEAT/" seguido del número de historia de usuario y "\_predev". Esta convención indica que la rama ha sido desarrollada por un miembro del equipo de Renovación de los Sistemas de Información.

3. Agregar los cambios realizados a la nueva rama y revisar cuidadosamente los archivos que se van a subir. Es importante verificar que no se incluyan archivos que solo se editaron en el entorno local, ya que esto puede causar problemas en la aplicación.

4. Antes de subir los cambios, escribir un mensaje de commit descriptivo. El mensaje debe comenzar con la palabra "FEAT" seguida del número de la historia de usuario entre paréntesis, seguido de dos puntos y un resumen conciso del desarrollo de la HU.

5. Hacer *push* de la rama al repositorio remoto.

6. Acceder a Gitlab, la herramienta de control de versiones utilizada por el equipo de Renovación de los Sistemas de Información. En Gitlab, generar un merge request, que es una solicitud para fusionar una rama de código con otra en el sistema de control de versiones. El objetivo del *merge request* es integrar los cambios realizados en una rama secundaria (por ejemplo, una rama de desarrollo) con una rama principal (por ejemplo, una rama de producción).

7. Asignar el *merge request* a otro desarrollador del equipo, quien revisará los cambios y verificará que todo esté correctamente desarrollado.

8. Una vez aprobado el *merge request*, se ejecutan los pipelines correspondientes. Un pipeline es una secuencia automatizada de pasos que se ejecutan de forma secuencial para construir, probar y desplegar el software de manera eficiente y consistente. Los *pipelines* del proyecto se definen a continuación:

- a. **Filter (Filtrado):** El *pipeline* de filtrado es el encargado de recibir los cambios realizados en el repositorio de código y aplicar una serie de criterios para determinar si

los cambios deben ser procesados y continuar con el flujo de trabajo. Por ejemplo, puede verificar si los cambios están en una rama específica, si cumplen con los requisitos de calidad o si están asociados a una determinada funcionalidad.

b. **Setup (Configuración):** El *pipeline* de configuración se encarga de establecer el entorno necesario para ejecutar los pasos siguientes. Esto implica la configuración de dependencias, la instalación de herramientas y la preparación de cualquier recurso o configuración adicional requerida para el proceso de construcción y prueba.

c. **Build (Construcción):** El *pipeline* de construcción se encarga de compilar el código fuente y generar los artefactos necesarios para la aplicación. Esto implica la compilación de los archivos fuente, la gestión de dependencias, la creación de paquetes binarios u otros formatos de distribución, y la generación de cualquier recurso necesario para la ejecución de la aplicación.

d. **Test (Pruebas):** El *pipeline* de pruebas se utiliza para ejecutar diferentes tipos de pruebas en el código construido. Esto puede incluir pruebas unitarias, pruebas de integración, pruebas de rendimiento, pruebas de seguridad, entre otras. El objetivo es garantizar que el software funcione correctamente y cumpla con los criterios de calidad establecidos.

e. **Deploy (Despliegue):** El *pipeline* de despliegue se encarga de implementar y poner en funcionamiento la aplicación en el entorno objetivo, ya sea de prueba o producción. Esto implica la configuración de servidores, la transferencia de los artefactos generados en el paso de construcción, la configuración de variables de entorno y cualquier otra tarea necesaria para poner en marcha la aplicación.

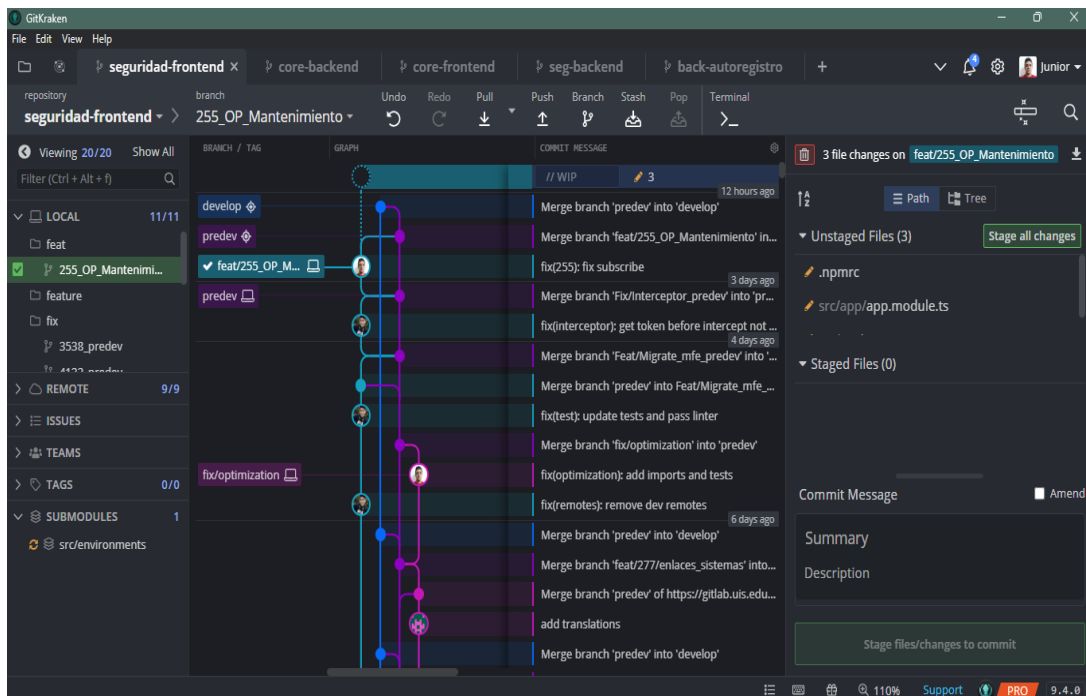
f. **Notification (Notificación):** se encarga de enviar notificaciones o alertas a los diferentes equipos o personas interesadas en el estado o resultados de un flujo de trabajo determinado.

Si alguno de estos *pipelines* falla, se analizará cual, y se realizará el debido proceso para su solución, si ese no es el caso, los cambios se fusionan en la rama objetivo, lo que permite que los nuevos desarrollos estén disponibles en la rama principal del proyecto.

Es importante seguir estos pasos de manera diligente para garantizar un flujo de trabajo ordenado y mantener la integridad y calidad del código en el repositorio.

### Figura 39.

*Interfaz de Gitkraken Client.*



## 7.3 Componentes Funcionales

### 7.3.1 Home de la Configuración General

**Figura 40.**

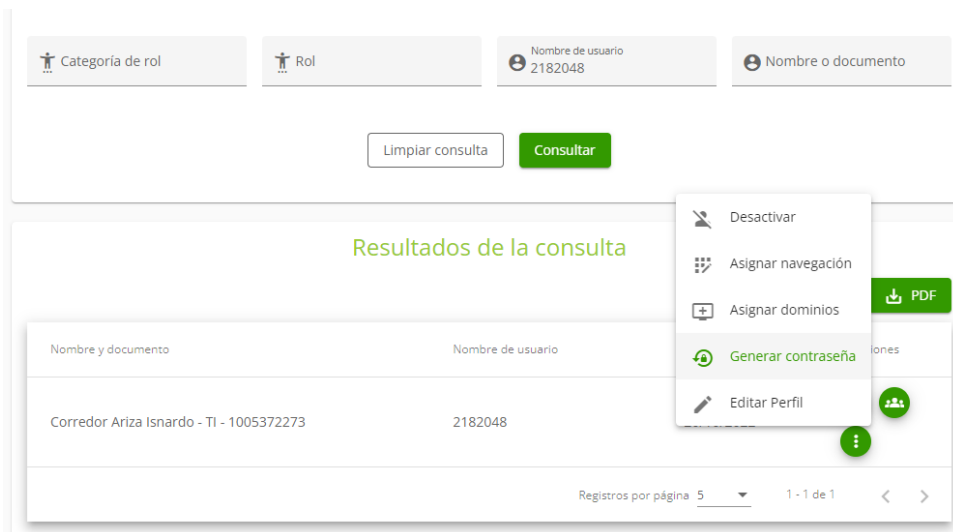
*Home del módulo de configuración general*



### 7.3.2 Acción de Generar Contraseña en el Listado de Usuarios

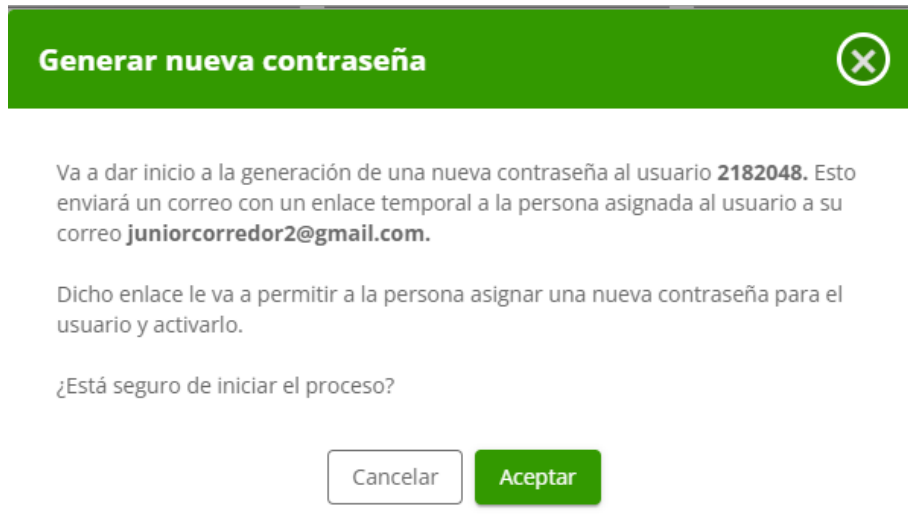
**Figura 41.**

*Interfaz del buscador de usuarios.*



**Figura 42.**

*Modal de la acción de generar nueva contraseña.*



**Generar nueva contraseña** ✕

Va a dar inicio a la generación de una nueva contraseña al usuario **2182048**. Esto enviará un correo con un enlace temporal a la persona asignada al usuario a su correo **juniorcorredor2@gmail.com**.

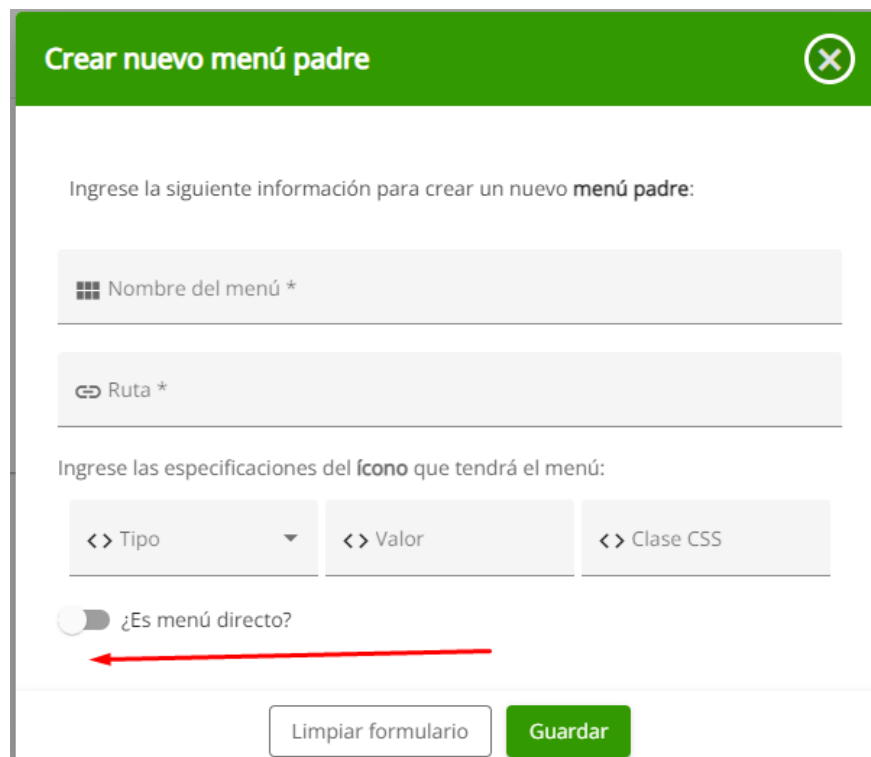
Dicho enlace le va a permitir a la persona asignar una nueva contraseña para el usuario y activarlo.

¿Está seguro de iniciar el proceso?

### 7.3.3 Pintar Menús de Acceso Directo en la Barra Superior Derecha

**Figura 43.**

*Toogle para habilitar el acceso directo de un menú.*



**Crear nuevo menú padre** ✕

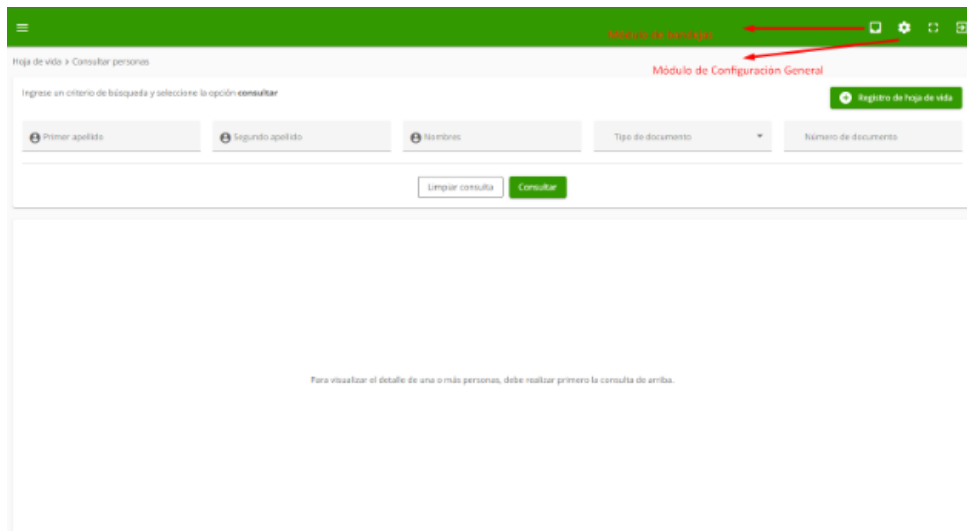
Ingrese la siguiente información para crear un nuevo **menú padre**:

Ingrese las especificaciones del **ícono** que tendrá el menú:

¿Es menú directo?

**Figura 44.**

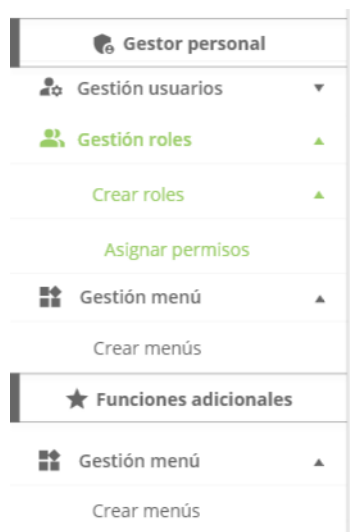
*Íconos de menús de acceso directo.*



### 7.3.4 Zonificar el Menú de Navegación Lateral Izquierdo

**Figura 45.**

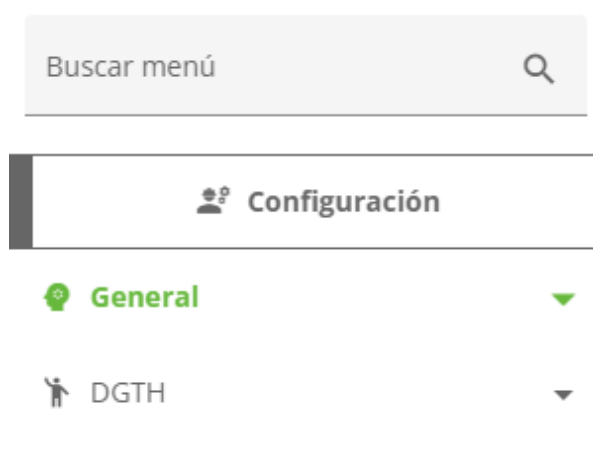
*Zonificación del menú lateral izquierdo.*



### 7.3.5 Buscador en el Menú de Tercer Nivel

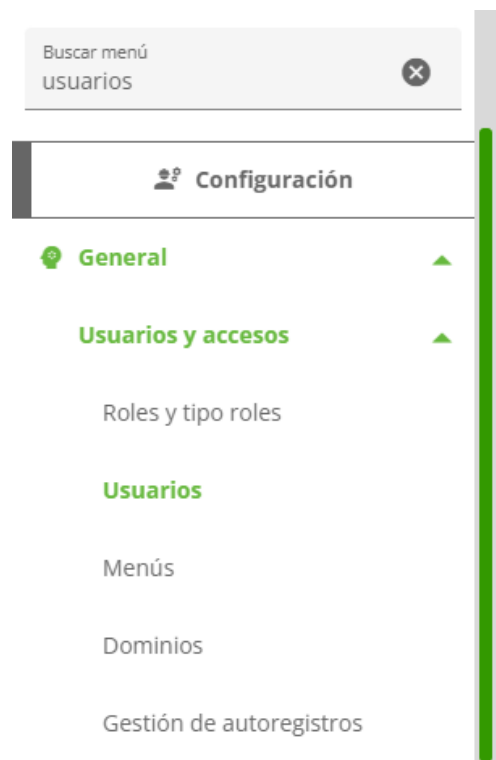
**Figura 46.**

*Input del buscador de menús.*



**Figura 47.**

*Buscador de menús en función.*

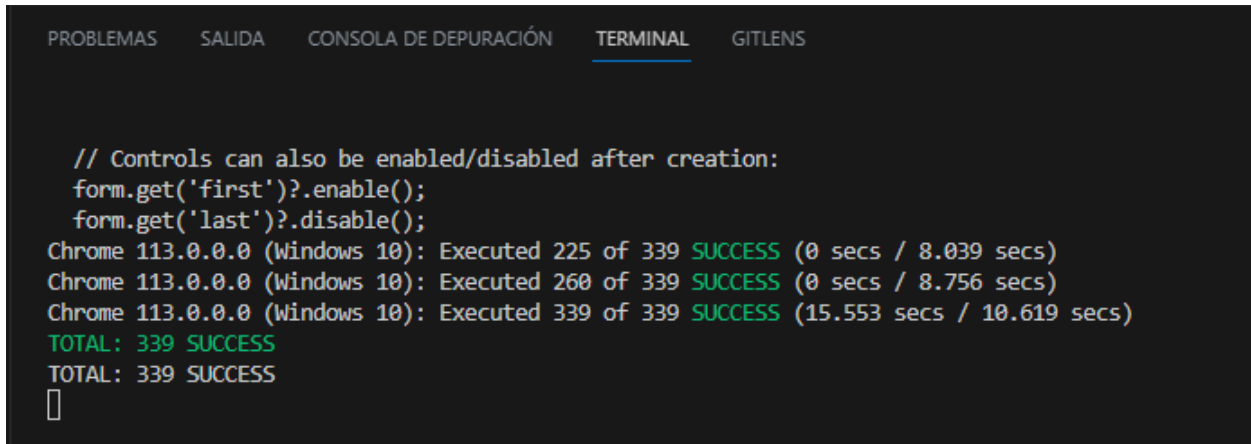


Como función adicional, el botón con forma de ícono cancel reinicia la búsqueda y acomoda el árbol de menús a su estado inicial, es decir, como en la imagen 28.

### 7.3.6 Estabilización de Test de Frontend

#### Figura 48.

Salida de Consola del módulo CORE.

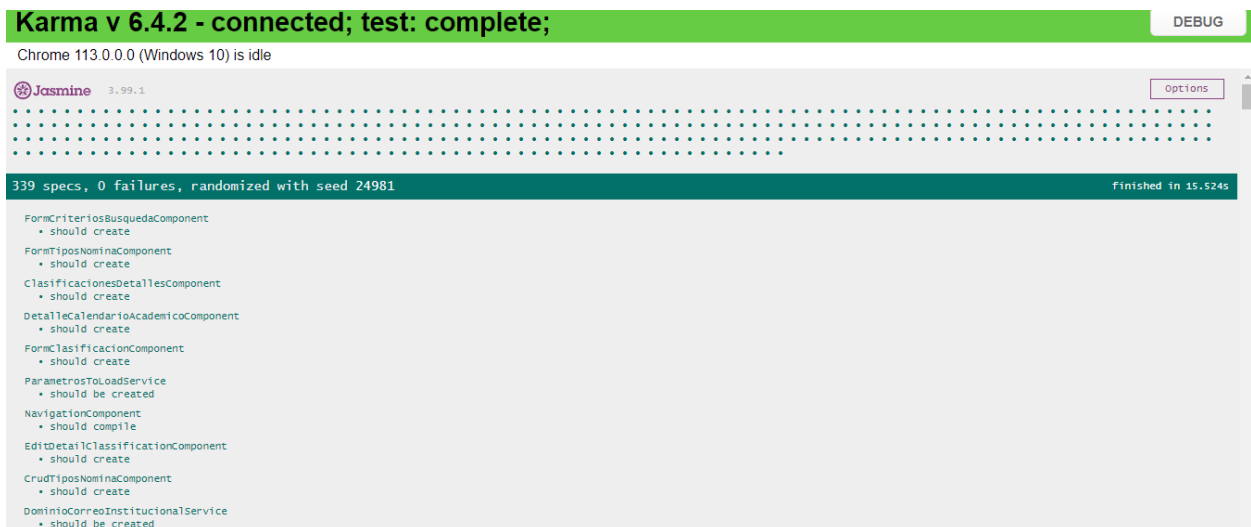


```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  GITLENS

// Controls can also be enabled/disabled after creation:
form.get('first')?.enable();
form.get('last')?.disable();
Chrome 113.0.0.0 (Windows 10): Executed 225 of 339 SUCCESS (0 secs / 8.039 secs)
Chrome 113.0.0.0 (Windows 10): Executed 260 of 339 SUCCESS (0 secs / 8.756 secs)
Chrome 113.0.0.0 (Windows 10): Executed 339 of 339 SUCCESS (15.553 secs / 10.619 secs)
TOTAL: 339 SUCCESS
TOTAL: 339 SUCCESS
█
```

#### Figura 49.

Salida de consola en el navegador del módulo de CORE.

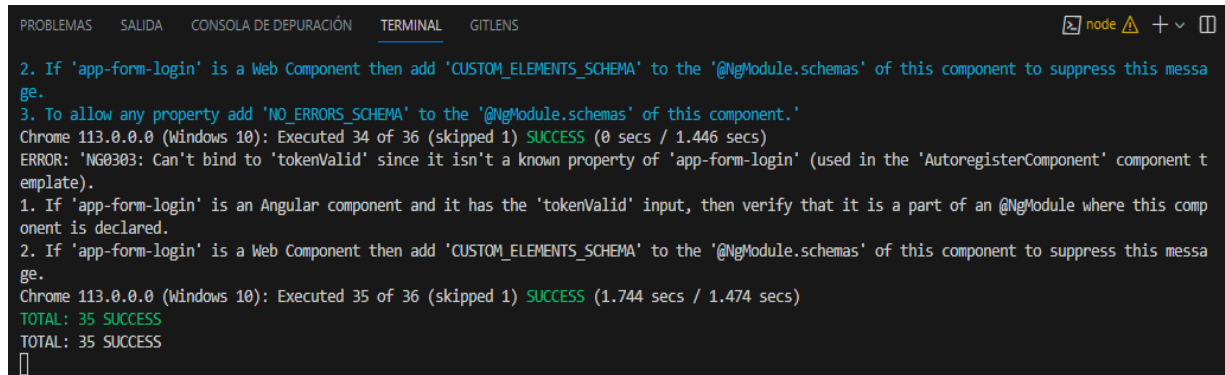


```
Karma v 6.4.2 - connected; test: complete;  DEBUG
Chrome 113.0.0.0 (Windows 10) is idle
Jasmine 3.99.1 Options
.....
339 specs, 0 failures, randomized with seed 24981  finished in 15.524s

FormCriteriosBusquedaComponent
  • should create
FormTiposNominaComponent
  • should create
ClasificacionesDetallesComponent
  • should create
DetalleCalendarioAcademicoComponent
  • should create
FormClasificacionComponent
  • should create
ParametrosToLoadService
  • should be created
NavigationComponent
  • should compile
EditDetalleClasificacionComponent
  • should create
CrudTiposNominaComponent
  • should create
DominioCorreoInstitucionalService
  • should be created
```

**Figura 50.**

*Salida de Consola del módulo AUTHORIZATION.*

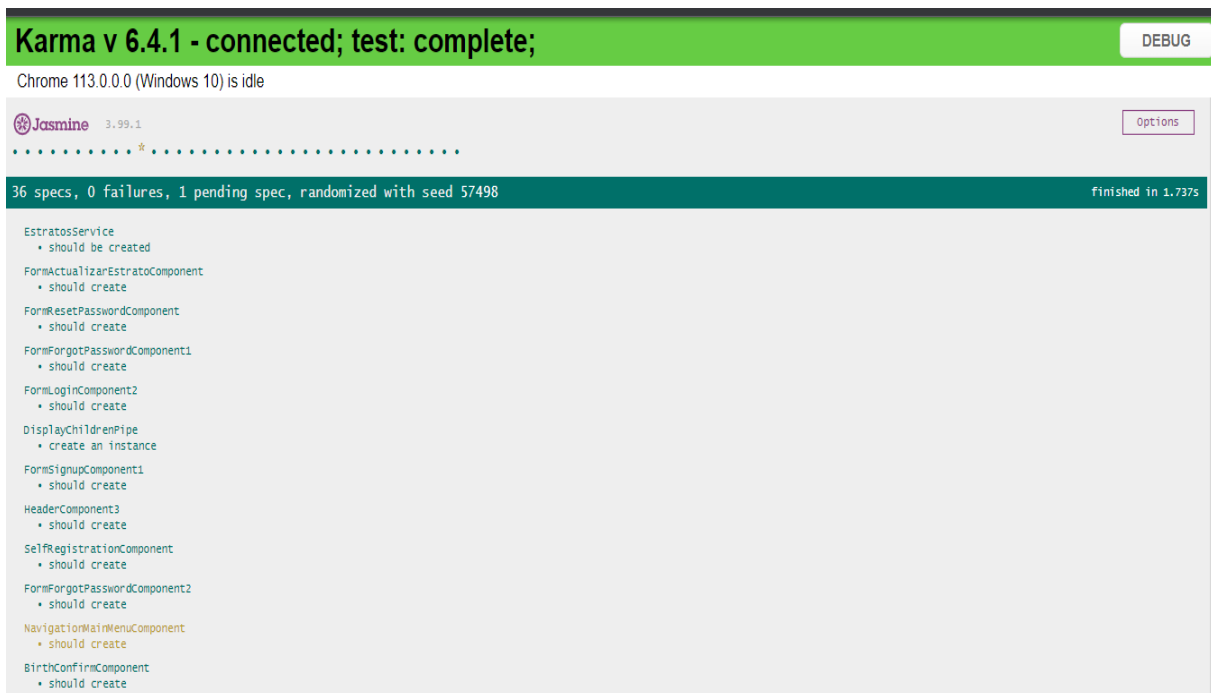


```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  GITLENS  node  +  ▢

2. If 'app-form-login' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the '@NgModule.schemas' of this component to suppress this message.
3. To allow any property add 'NO_ERRORS_SCHEMA' to the '@NgModule.schemas' of this component.'
Chrome 113.0.0.0 (Windows 10): Executed 34 of 36 (skipped 1) SUCCESS (0 secs / 1.446 secs)
ERROR: 'NG0303: Can't bind to 'tokenValid' since it isn't a known property of 'app-form-login' (used in the 'AutoregisterComponent' component template).
1. If 'app-form-login' is an Angular component and it has the 'tokenValid' input, then verify that it is a part of an @NgModule where this component is declared.
2. If 'app-form-login' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the '@NgModule.schemas' of this component to suppress this message.
Chrome 113.0.0.0 (Windows 10): Executed 35 of 36 (skipped 1) SUCCESS (1.744 secs / 1.474 secs)
TOTAL: 35 SUCCESS
TOTAL: 35 SUCCESS
█
```

**Figura 51.**

*Salida de consola en el navegador del módulo de AUTHORIZATION.*



```
Karma v 6.4.1 - connected; test: complete;  DEBUG
Chrome 113.0.0.0 (Windows 10) is idle
Jasmine 3.99.1
.....*.....
36 specs, 0 failures, 1 pending spec, randomized with seed 57498  Finished in 1.737s

EstratosService
  • should be created
FormActualizarEstratoComponent
  • should create
FormResetPasswordComponent
  • should create
FormForgotPasswordComponent1
  • should create
FormLoginComponent2
  • should create
DisplayChildrenPipe
  • create an instance
FormSignupComponent1
  • should create
HeaderComponent3
  • should create
SelfRegistrationComponent
  • should create
FormForgotPasswordComponent2
  • should create
NavigationMainMenuComponent
  • should create
BirthConfirmComponent
  • should create
```

### 7.3.7 Crud de Autoregistros

**Figura 52.**

*CRUD de autoregistro.*

Gestión de autoregistros				
Nombre autoregistro	Descripción	Fecha desde	Fecha hasta	Acciones
Autoregistro UIS	Autoregistro para acceso de nuevas personas contra...	19/01/2023		  
Developers rsi	Permite que los desarrolladores obtengan el rol de...	15/05/2023		  
Aspirantes cátedra	Permite iniciar el proceso como aspirante a convoc...	02/09/2022		  

Registros por página 5 1 - 3 de 3 < >

**Figura 53.**

*Crear autoregistro.*

**Crear autoregistro**
✕

📄 Nombre del autoregistro \*

Observaciones \*

0 de 400 caracteres


Fecha desde \*  
15/05/2023


Fecha hasta


**Esquema general** i

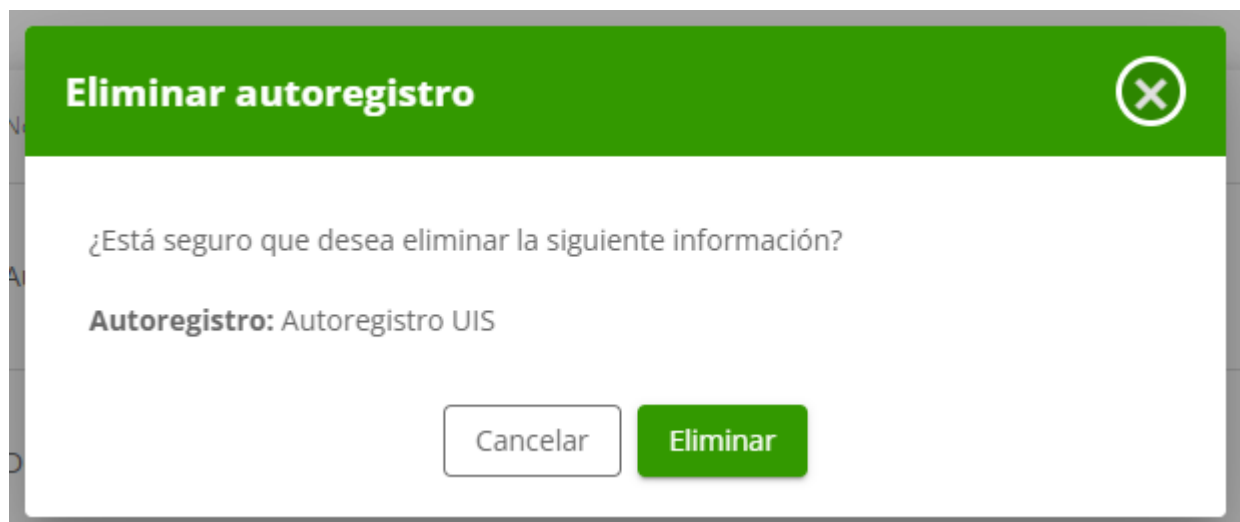
Limpiar formulario

Guardar

**Figura 54.***Editar autoregistro.*

The screenshot shows a form titled "Editar autoregistro" with a close button in the top right corner. The form contains the following fields and controls:

- Nombre del autoregistro \***: A text input field containing "Autoregistro UIS".
- Observaciones \***: A text area containing "Autoregistro para acceso de nuevas personas contratadas de la UIS." with a character count of "66 de 400 caracteres".
- Fecha desde \***: A date picker field showing "19/01/2023".
- Fecha hasta**: An empty date picker field.
- Esquema general**: A checked checkbox with an information icon.
- Buttons**: "Cancelar" and "Guardar" buttons at the bottom.

**Figura 55.***Eliminar autoregistro.*

The screenshot shows a confirmation dialog titled "Eliminar autoregistro" with a close button in the top right corner. The dialog contains the following text and controls:

- Message**: "¿Está seguro que desea eliminar la siguiente información?"
- Item Name**: "Autoregistro: Autoregistro UIS"
- Buttons**: "Cancelar" and "Eliminar" buttons at the bottom.

## 8. Validación

Para el seguimiento y control, se consideran los casos de prueba aplicables a la historia de usuario (HU), así como una lista de chequeo basada en los estándares establecidos por el equipo de UX y las secciones de prueba analizadas por QA. Estos mecanismos se utilizan para verificar el cumplimiento y funcionamiento adecuados de la HU entregada.

Los casos de prueba están compuestos de unos ítems utilizados para verificar el correcto funcionamiento de la HU, estos son:

- Id: El id del caso de prueba.
- Rol: El rol con el que se hace la prueba.
- La descripción de la prueba: nombre corto de la prueba y descripción. Se inicia con un verbo en infinitivo.
- Precondición: componentes que deben estar listos antes del inicio de la prueba.
- Pasos: paso a paso a realizar para el caso de prueba.
- Resultado esperado: cada resultado esperado debe mostrar el comportamiento exitoso de prueba.
- Set de datos: se relacionan variables y datos utilizados de la prueba.
- Ciclo: estados definidos.
- Observaciones: anotaciones realizadas por QA.

La lista de chequeo general aplica los siguientes ítems para la verificación de los estándares y funcionalidades:

- Sección: aquí se especifica la sección a la cual se aplica el estándar.

- Descripción de la prueba: nombre corto de la prueba y descripción del objetivo de la prueba.
- Resultado esperado: cada resultado debe mostrar el comportamiento exitoso de la prueba.
- Columna de Aplicación: verifica si aplica o no el estándar.
- Ciclo: estados definidos.
- Observaciones: anotaciones realizadas por QA.

## 8.1 Pruebas para la Home de Configuración general

**Tabla 1.**

*Pruebas para la Home de Configuración general - Casos de prueba*

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
CP-HU-3310-001	Administrador	Validar el direccionamiento a la página principal del menú de configuración.	Usuario con rol Administrador .	1. Ingrese con rol Administrador al menú principal 2. Seleccione el menú Configuración	Al ingresar al menú Configuración siendo Administrador, se redirecciona a la página principal de bienvenida al menú.	N/A.	Cumple	La página principal se plantea a mostrar a los "diferentes tipos de administradores" lo cual hace referencia a usuarios de otras secciones diferentes a la administración de la plataforma

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
								que tienen acceso a ciertos módulos del menú Configuración.
CP-HU-3310-002	Administrador	Validar el ajuste de criterios establecidos en mockup para la página principal del menú.	Usuario con rol Administrador .	1. Ingrese con rol Administrador al menú principal 2. Seleccione el menú Configuración	La página principal del menú se ajusta al diseño establecido con el ícono propio del menú y los mensajes indicativos especificados.	N/A.	Cumple .	No se registraron observaciones.
CP-HU-3310-003	Administrador	Validar la redirección a la	Usuario con rol	1. Ingrese con rol	Al intentar ingresar a un	N/A.	N/A.	Esta funcionalidad no

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
		página principal del menú al usar una URL inválida.	Administrador	Administrador al menú principal 2. Seleccione el menú Configuración	módulo que haga parte del menú Configuración con una URL inválida (mal escrita o inexistente) se redirecciona al usuario a la página principal del menú e indica que no existe tal módulo.			está requerida/especificada, pero se presenta como sugerencia para que exista un comportamiento más dinámico en el sistema.

**Tabla 2.***Pruebas para la Home de Configuración general - Lista de chequeo*

Sección	Descripción de la prueba	Resultado esperado	Aplica S/N	Ciclo	Observaciones
Responsive - Dispositivo móviles	"Validar que los modales y menus correspondientes se adecuen a dispositivos moviles:  - Alineación  - Botones  - Listados  Tener en cuenta la dimensión mínima 360 "	N/A	S	Cumple	No se registran observaciones

## 8.2 Pruebas para Acción de Generar Nueva Contraseña en Listado de Usuarios

**Tabla 3.**

*Pruebas para Acción de Generar Nueva Contraseña en Listado de Usuarios - Casos de prueba*

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
CP-HU-3423-001	Administrador	Verificar el funcionamiento del botón de generación de contraseña para un usuario.	Usuario existente en la plataforma.	1. Ingrese con rol Administrador 2. Diríjase a Configuración/General/Usuarios y accesos/Usuarios 3. Realice la consulta de usuario a generar contraseña 4. Seleccione el botón de acciones (3 puntos) en el	La nueva acción despliega un modal para el proceso de generación de contraseña a un usuario.	N/A	Cumple	No se registran observaciones

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
				registro del usuario				
				5. Seleccione la opción "Generar contraseña"				
CP-HU-3423-002	Administrador	Validar el mensaje de información de cambio de contraseña para un usuario	Usuario existente en la plataforma.	<p>1. Ingrese con rol Administrador</p> <p>2. Diríjase a Configuración/General/Usuarios y accesos/Usuarios</p> <p>3. Realice la consulta de usuario a generar contraseña</p> <p>4. Seleccione el botón de acciones</p>	<p>Al seleccionar, se muestra un modal con mensaje especificado en la HU, el cual informa del proceso de restablecimiento y desactivación</p>	Nombre de usuario y Correo electrónico de usuario	Cumple	Se cumple el requisito, pero se debe hacer un ajuste al texto del mensaje ya que se descarta la funcionalidad de desactivación del usuario al realizar el proceso de

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
				(3 puntos) en el registro del usuario	del registro de usuario			generación de contraseña.
				5. Seleccione la opción "Generar contraseña"				Revisado para Ciclo 2 luego de haber hecho el cambio y se cumple.
CP-HU-3423-003	Administrador	Validar el mensaje de confirmación al generar una nueva contraseña a un usuario	Usuario existente en la plataforma.	1. Ingrese con rol Administrador 2. Diríjase a Configuración/General/Usuarios y accesos/Usuarios 3. Realice la consulta de	Al realizar el proceso de generación de nueva contraseña, se muestra un mensaje emergente confirmando	N/A	Cumple	El mensaje de confirmación debe indicar el envío de correo al usuario sobre la desactivación de este. Al cambiar parte del flujo de la

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
				usuario a generar contraseña	que el proceso se realizó correctamente.			funcionalidad removiendo la desactivación de usuario, el mensaje de confirmación se debe mantener respecto al envío de correo.
				4. Seleccione el botón de acciones (3 puntos) en el registro del usuario				Revisado en ciclo 2 luego de hacer el cambio y se cumple.
				5. Seleccione la opción "Generar contraseña"				
CP-HU-3423-004	Administrador	Validar el envío de correo para generación de	Usuario existente en la plataforma.	1. Ingrese con rol Administrador	Al realizar el proceso de generación de	N/A	Cumple	No se registran observaciones

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
		contraseña a un usuario		<p>2. Diríjase a Configuración/General/Usuarios y accesos/Usuarios</p> <p>3. Realice la consulta de usuario a generar contraseña</p> <p>4. Seleccione el botón de acciones (3 puntos) en el registro del usuario</p> <p>5. Seleccione la opción "Generar contraseña"</p> <p>6. En el modal emergente,</p>	<p>nueva contraseña a un administrador, este recibe un correo electrónico referente al cambio de contraseña.</p>			

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
				seleccione				
				"Aceptar"				
CP-HU-3423-005	Administrador	Validar el envío de correo para generación de contraseña a correo institucional	Usuario existente en la plataforma.	1. Ingrese con rol Administrador 2. Diríjase a Configuración/General/Usuarios y accesos/Usuarios 3. Realice la consulta de usuario a generar contraseña 4. Seleccione el botón de acciones (3 puntos) en el registro del usuario	Al realizar el proceso de generación de nueva contraseña a un usuario que no tenga correo electrónico personal registrado, el proceso de envío de correo de restablecimien	N/A	Cumple	No se registran observaciones

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
				5. Seleccione la opción "Generar contraseña"	to debe hacerse y enviarse al correo			
				6. En el modal emergente, seleccione "Aceptar"	institucional del usuario.			

**Tabla 4.**

*Pruebas para Acción de Generar Nueva Contraseña en Listado de Usuarios - Lista de chequeo*

Sección	Descripción de la prueba	Resultado esperado	Aplica S/N	Ciclo	Observaciones
Responsive - Dispositivo móviles	"Validar que los modales y menus correspondientes se adecuen a dispositivos moviles: - Alineación - Botones - Listados Tener en cuenta la dimensión mínima 360 "	N/A	S	Cumple	No se registran observaciones

### 8.3 Pruebas para Pintar Menús de Acceso Directo en la Barra Superior Derecha

**Tabla 5.**

*Pruebas para Pintar Menús de Acceso Directo en la Barra Superior Derecha - Casos de prueba*

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
CP-HU-3354-001	Cualquiera	Verificar la visualización como losetas de menús que conforman el menú superior	Menús con la opción "Menú superior" activa	1. Ingrese con cualquier rol que tenga asignados menús pertenecientes al menú superior 2. Seleccione e ingrese a cualquier módulo de la plataforma	Al estar en cualquier módulo del SIA, el menú superior derecho se visualiza con los íconos de los menús del sistema que también son losetas del menú principal	N/A	Cump	Existen menús que no necesariamente son losetas y aun así hacen parte del menú superior, lo cual depende totalmente de la administración actual de los

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
								menús del sistema.
CP-HU-3354-002	Administrador	Verificar el orden de posicionamiento de los menús	Menús con la opción "Menú superior" activa	<ol style="list-style-type: none"> <li>Ingrese con rol Administrador</li> <li>Seleccione e ingrese a cualquier módulo de la plataforma</li> <li>Diríjase al menú superior</li> </ol>	El orden de posicionamiento de los menús (íconos) debe obedecer al orden establecido por ID	N/A	Cumple	El orden de aparición de los menús asignados está en proceso de mejora continua. Aunque se plantea que el orden debe seguir la secuencia de los menús en la plataforma,

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
								algunos de ellos no tienen una posición de orden específica definida. Se está trabajando para garantizar que, en ausencia de un ID de orden específico, se organicen alfabéticamente.

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
								te de manera consistente y efectiva.
CP-HU-3354-003	Administrador	Validar que en la visualización de menús no existan aquellos que hacen parte del módulo actual	Menús con la opción "Menú superior" activa	<ol style="list-style-type: none"> <li>Ingrese con rol Administrador</li> <li>Seleccione e ingrese a cualquier módulo de la plataforma</li> <li>Diríjase al menú superior</li> </ol>	En el menú superior, los menús que se muestran asignados deben ser aquellos que pertenezcan únicamente a módulos diferentes al que el usuario se encuentra navegando.	N/A	Cumple	Los menús que apuntan al módulo en el que se está navegando no deben aparecer en el menú superior dependiendo del módulo en el que se esté en el momento.

<b>ID</b>	<b>Rol</b>	<b>Descripción de la prueba</b>	<b>Precondición</b>	<b>Pasos</b>	<b>Resultado esperado</b>	<b>Set de datos</b>	<b>Ciclo</b>	<b>Observaciones</b>
CP-HU-3354-004	Administrador	Validar que el acceso a menús del menú superior se realice en pestañas nuevas del navegador	Menús con la opción "Menú superior" activa	1. Ingrese con rol Administrador 2. Seleccione e ingrese a cualquier módulo de la plataforma 3. Diríjase al menú superior 4. Seleccione algún menú habilitado	Al acceder a un menú del menú superior, este debe abrirse en una nueva pestaña en lugar de la misma, de tal manera que el usuario no pierda la navegación actual.	N/A	Cumple	No se registraron observaciones
CP-HU-3354-005	Administrador	Validar que el acceso a menús por losetas se realice en la misma	Menús con la opción "Menú superior" activa	1. Ingrese con rol Administrador 2. Seleccione e ingrese a	Al acceder a un menú por medio de la interfaz de losetas, este	N/A	Cumple	No se registraron observaciones

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
		pestaña de navegación		cualquier módulo de la plataforma por medio de las losetas	debe abrirse en la misma pestaña de navegación que se encuentra el usuario.			
CP-HU-3354-006	Administrador	Validar que los menús asignados y mostrados en el menú superior pueden ser accedidos y abren sus respectivas interfaces	Menús con la opción "Menú superior" activa	<ol style="list-style-type: none"> <li>Ingrese con rol Administrador</li> <li>Seleccione e ingrese a cualquier módulo de la plataforma</li> <li>Diríjase al menú superior</li> </ol>	Al acceder a un menú superior, se abre una nueva pestaña con la interfaz propia del menú seleccionado.	N/A	No cumple	<p>Cuando se asignan menús padres que funcionan como contenedores sin tener una pantalla específica asociada, se</p>

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
				4. Seleccione algún menú habilitado				ha identificado un problema en la redirección correspondiente. En algunos casos, no se está estableciendo la correcta asociación con los submenús principales que han sido asignados a

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
								dichos menús padres. Esta situación ha llevado a que se acepte la Historia de Usuario (HU) sin contemplar este error, lo cual ha generado la apertura de una nueva HU con las modificacione s necesarias.

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
								La nueva HU ha sido asignada a otro desarrollador para su implementación.
CP-HU-3354-007	Administrador	Verificar la funcionalidad de activación de un menú como menú superior al crearlo	Menú en proceso de creación	1. Ingrese con rol Administrador 2. Diríjase a Configuración>Usuarios y accesos>Menús	Al crear un nuevo menú con la opción "Menú superior derecho" activa, este se agregará automáticamente a la sección del	N/A	Cumple	No se registraron observaciones.

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
				3. Seleccione la opción "Crear menú"	menú superior para ser accedido desde allí.			
				4. En la creación del menú, active la opción "Menú superior derecho"				
CP-HU-3354-008	Administrador	Verificar la funcionalidad de activación de un menú como menú superior al editarlo	Menú ya existente en la plataforma	1. Ingrese con rol Administrador 2. Diríjase a Configuración>Usuarios y accesos>Menús 3. Seleccione el menú a editar	Al editar un menú activando la opción "Menú superior", este se agregará automáticamente a la sección del menú superior	N/A	No cumple	En algunos menús, al seleccionar la opción para añadirlos al menú superior se está desplegando un mensaje de

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
				4. En la edición del menú, active la opción "Menú superior derecho"	para ser accedido desde allí.			error "Ya existe un menú con este id". Pareciera ser un error a partir de menús que comparten el mismo nombre y son menús padres. Esta situación ha llevado a que se acepte la Historia de Usuario (HU) sin

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
								contemplar este error, lo cual ha generado la apertura de una nueva HU con las modificaciones necesarias. La nueva HU ha sido asignada a otro desarrollador para su implementación.

**Tabla 6.***Pruebas para Pintar Menús de Acceso Directo en la Barra Superior Derecha - Lista de chequeo*

<b>Sección</b>	<b>Descripción de la prueba</b>	<b>Resultado esperado</b>	<b>Aplica S/N</b>	<b>Ciclo</b>	<b>Observaciones</b>
Formularios	Validar que el modal relacionado en la HU coincida con los campos relacionados en pantalla	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que los nombres de los campos se encuentren en modo oración	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que todos los botones de acción (Sólo si aplican dentro de un formulario tengan su respectivo Tooltip)	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que todos los campos del formulario tengan el icono en el costado izquierdo, excepto para el icono del calendario.	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que en los campos de listas desplegables y registro de información tengan sus respectivos textos indicativos.	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que al dejar vacíos campos obligatorios del formulario el sistema no permita guardar información y muestre mensaje de error - modo creación y edición	N/A	S	Cumple	No se registran observaciones

Sección	Descripción de la prueba	Resultado esperado	Aplica S/N	Ciclo	Observaciones
Formularios	Validar que el sistema no permita ingresar registros duplicados (Debe especificarse en la HU) Ejemplo: Crear un permiso con los mismos datos de un registro ya creado)	N/A	S	Cumple	No se registran observaciones
Formularios	"Validar que al ejecutar la opción de Limpiar formulario el sistema elimine la información ingresada.  1. Para el caso del radio button estos deben aparecer con la misma selección que tenían antes de la modificación.  2. Para el caso de las fechas donde por defecto está marcada la fecha actual, esta debe borrarse al limpiar formulario."	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que la pantalla de consulta cumpla con los datos definidos en la HU y los formatos para datos estándar estén definidos correctamente	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que todos los botones de acción tengan su respectivo Tooltip	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que el botón de cancelar y la (X) del modal, cierren la pantalla correspondiente.	N/A	S	Cumple	No se registran observaciones

Sección	Descripción de la prueba	Resultado esperado	Aplica S/N	Ciclo	Observaciones
Formularios	Validar que, al actualizar un registro, este se mantenga en la misma posición en la que se encontraba antes de su edición en el listado de registros.	N/A	S	Cumple	No se registran observaciones
Formularios	"Validar que los modales y menús correspondientes se adecuen a dispositivos móviles:  - Alineación  - Botones  - Listados   Tener en cuenta la dimensión mínima 360 "	N/A	S	Cumple	No se registran observaciones

## 8.4 Pruebas para Zonificar el Menú de Navegación Vertical Izquierdo

**Tabla 7.**

*Pruebas para Zonificar el Menú de Navegación Vertical Izquierdo - Casos de prueba*

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo I	Observaciones
CP-HU-3369-001	Cualquiera - administrador	Verificar la visualización de menús asignados a un usuario como funciones adicionales	El usuario debe tener uno o más menús del módulo a navegar adicionales a los asignados al rol que usa.	<ol style="list-style-type: none"> <li>Ingrese a cualquier módulo del SIA</li> <li>Identifique el menú lateral izquierdo</li> <li>Identifique las secciones divididas de menús asignados.</li> </ol>	<p>Cuando el usuario tiene menús adicionales a los asignados al rol que usa en el módulo que se encuentra navegando (menús asignados al usuario), el menú lateral debe contener</p>	N/A	Cumple	No se registran observaciones

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo I	Observaciones
					una nueva sección de "Funciones adicionales" que contiene dichos menús asignados al usuario.			
CP-HU-3369-002	Cualquiera - administrador	Validar la búsqueda en dos secciones del menú lateral	El usuario debe tener uno o más menús del módulo a navegar adicionales a los asignados al rol que usa.	1. Ingrese a cualquier módulo del SIA 2. Identifique el menú lateral izquierdo 3. Seleccione el buscador del menú lateral	Al realizar una búsqueda en el menú lateral, esta se debe realizar simultáneamente en la sección de "Funciones adicionales"	Nombre de menú a buscar	Cumple	No se registran observaciones

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo I	Observaciones
				4. Realice la búsqueda de menú.	para filtrar coincidencias allí.			
CP-HU-3369-003	Cualquiera - administrador	Verificar la visualización de scroll lateral en el menú lateral	El usuario debe tener uno o más menús del módulo a navegar adicionales a los asignados al rol que usa.	1. Ingrese a cualquier módulo del SIA 2. Identifique el menú lateral izquierdo 3. Identifique las secciones divididas de menús asignados.	Al navegar por el menú lateral izquierdo, se debe mostrar un scroll horizontal al tener menús en niveles de profundidad que no sean posibles de visualizar de manera completa	N/A	Cumple	El scroll vertical se visualiza al tener diferentes niveles de visualización. Se implementa una estructuración del árbol de menús tal que no se muestre un scroll horizontal y se

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo I	Observaciones
								ubiquen de mejor manera los nombres de los menús. Probado en ciclo 2 con mejor desempeño.
CP-HU-3369-004	Cualquiera - administrador	Verificar la no repetición de menús en la sección de Funciones adicionales para un usuario	El usuario debe tener uno o más menús del módulo a navegar adicionales a los asignados al rol que usa.	1. Ingrese a cualquier módulo del SIA 2. Identifique el menú lateral izquierdo 3. Identifique las secciones divididas de	Al tener menús asignados adicionales a los asignados al rol del usuario, la sección "Funciones adicionales" del panel lateral no	N/A	Cumple	No se registran observaciones

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo I	Observaciones
				menús asignados.	debe contener menús repetidos a los que se visualizan en la sección principal, a menos que contenga menús que hacen parte de menús contenedores, lo que haría que se muestre un duplicado del menú contenedor con el menú asignado			

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo I	Observaciones
					adicional al usuario.			

**Tabla 8.***Pruebas para Zonificar el Menú de Navegación Vertical Izquierdo - Lista de chequeo*

Sección	Descripción de la prueba	Resultado esperado	Aplica S/N	Ciclo	Observaciones
Responsive Dispositivo móviles	- "Validar que los modales y menús correspondientes se adecuen a dispositivos móviles:  - Alineación  - Botones  - Listados  Tener en cuenta la dimensión mínima 360 "	N/A	S	Cumple	No se registran observaciones

## 8.5 Pruebas para el Buscador en el Menú de Tercer Nivel

Tabla 9.

*Pruebas para el Buscador en el Menú de Tercer Nivel - Casos de prueba*

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
CP-HU-3317-001	Usuario	Verificar que, al utilizar el buscador de menú de tercer nivel, se filtre correctamente y muestre las opciones relevantes al usuario.	El usuario debe tener acceso al menú y haber ingresado al buscador.	1. Ingrese al menú principal. 2. Acceda al buscador de menú de tercer nivel. 3. Ingrese una cadena de caracteres correspondiente a una opción específica del menú.	El buscador debe mostrar solo las opciones del menú que coinciden con la cadena de caracteres ingresada, sin importar mayúsculas o minúsculas. Las opciones deben estar	N/A	Cumple	Verificar que el resultado del filtro sea correcto y muestre las opciones correspondientes al usuario.

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
					relacionadas con el rol y permisos del usuario.			
CP-HU-3317-002	Administrador	Confirmar que, al utilizar el buscador de menú de tercer nivel, se resalten las opciones encontradas y se muestre la cadena de ancestros.	El administrador debe tener acceso al menú y haber ingresado al buscador.	1. Acceda al menú principal. 2. Utilice el buscador de menú de tercer nivel. 3. Ingrese una cadena de caracteres correspondiente a una opción específica del menú.	El buscador debe mostrar las opciones que coinciden con la cadena de caracteres ingresada. La opción seleccionada debe resaltarse en el color utilizado por el sistema y mostrar la	N/A	Cumple	Verificar que el resultado y la visualización de la cadena de ancestros sean correctos y brinden una navegación clara al administrador.

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
					cadena de			
					ancestros hasta			
					el tercer nivel			
					para facilitar la			
					navegación.			

**Tabla 10.***Pruebas para el Buscador en el Menú de Tercer Nivel - Lista de chequeo*

Sección	Descripción de la prueba	Resultado esperado	Aplica S/N	Ciclo	Observaciones
Responsive - Dispositivo móviles	"Validar que los modales y menús correspondientes se adecuen a dispositivos móviles:  - Alineación  - Botones  - Listados  Tener en cuenta la dimensión mínima 360 "	N/A	S	Cumple	No se registran observaciones

## 8.6 Pruebas para Estabilización de Test de Frontend

Tabla 11.

*Pruebas para Estabilización de Test de Frontend - Casos de prueba*

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
CP-HU-5513-001	Equipo de operaciones encargado del desarrollo de los módulos "Core" y "Authorization"	Verificar que los módulos "Core" y "Authorization" pasen los test unitarios utilizando las herramientas Jasmine y Karma.	El equipo de operaciones tiene acceso y permisos necesarios para realizar las pruebas unitarias y está familiarizado con Jasmine y Karma.	1. Ejecutar los test unitarios de los módulos "Core" y "Authorization" utilizando Jasmine y Karma.	Los test unitarios de los módulos "Core" y "Authorization" se ejecutan correctamente sin errores, indicando que los módulos están estabilizados y pasan los test unitarios.	N/A	Cumple	No se registraron observaciones.

<b>ID</b>	<b>Rol</b>	<b>Descripción de la prueba</b>	<b>Precondición</b>	<b>Pasos</b>	<b>Resultado esperado</b>	<b>Set de datos</b>	<b>Ciclo</b>	<b>Observaciones</b>
CP-HU-5513-002	Equipo de operaciones encargado del desarrollo de los módulos "Core" y "Authorization"	Verificar que se creen y ejecuten los test unitarios para garantizar el correcto funcionamiento del código.	El equipo de operaciones está familiarizado con la creación y ejecución de test unitarios.	1. Crear test unitarios para las funcionalidades clave de los módulos "Core" y "Authorization". 2. Ejecutar los test unitarios y verificar que todos los casos de prueba	Los test unitarios se crean de manera adecuada y todos los casos de prueba pasan, asegurando el correcto funcionamiento del código.	N/A	Cumple	No se registraron observaciones.

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
				pasen				
				correctame				
				nte.				

### Lista de chequeo

La aplicación de la lista de chequeo no es factible para esta historia de usuario, debido a su enfoque técnico en la estabilización del sistema. Se requieren herramientas y técnicas especializadas, como pruebas unitarias y revisión de código, para garantizar la estabilidad y calidad del sistema.

## 8.7 Pruebas para Crud de Autoregistros

**Tabla 12.**

*Pruebas para Crud de Autoregistros - Casos de prueba*

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
CP-HU-4791-001	Administrador	Verificar visualización de la tabla de autoregistros.	Interfaz de gestión de autoregistros creada.	<p>1. Iniciar sesión como administrador.</p> <p>2. Acceder a la sección de Gestión de autoregistros.</p> <p>3. Verificar la visualización de la tabla con las columnas correspondientes.</p>	<p>La tabla de autoregistros se muestra correctamente con las columnas especificadas.</p>	<p>Autoregistros previamente cargados en la base de datos.</p>	Cumple	<p>El orden de los autoregistros se basa en la fecha desde más menor registrada.</p>

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
CP-HU-4791-002	Administrador	Verificar la creación de un autoregistro.	No existen autoregistros previamente creados.	<p>1. Iniciar sesión como administrador.</p> <p>2. Acceder a la sección de Gestión de autoregistros.</p> <p>3. Crear un nuevo autoregistro completando los campos obligatorios con valores válidos.</p> <p>4. Guardar el autoregistro.</p>	El autoregistro se crea correctamente y se guarda en la base de datos.	Valores válidos para los campos obligatorios (Nombre, Descripción, fecha desde).	Cumple	Se debe verificar que se cumplan las restricciones de formato y validación de fechas especificadas en los requerimientos.
CP-HU-4791-003	Administrador	Verificar la edición de un autoregistro existente.	Existe al menos un autoregistro previamente	<p>1. Iniciar sesión como administrador.</p>	El autoregistro se edita correctamente y los cambios	Autoregistro previamente	Cumple	Se debe validar que los cambios realizados en los campos del

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
			creado en la plataforma.	2. Acceder a la sección de Gestión de autoregistros. 3. Seleccionar un autoregistro existente para editar.	se guardan en la base de datos.	creado en la plataforma.		autoregistro se reflejen correctamente después de guardar la edición.
				4. Modificar los campos deseados (por ejemplo, Nombre, Descripción, fecha desde, fecha hasta).				También se debe asegurar que se cumplan las restricciones y validaciones especificadas en los
				5. Guardar los cambios realizados.				requerimientos, como el formato y la relación entre fechas.

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
CP-HU-4791-004	Administrador	Verificar que no se pueda copiar un vínculo sin un rol asociado.	Existe un vínculo sin un rol asociado en la plataforma.	<p>1. Iniciar sesión como administrador.</p> <p>2. Acceder a la sección de Gestión de autoregistros.</p> <p>3. Identificar un vínculo sin un rol asociado.</p> <p>4. Intentar copiar el vínculo.</p>	<p>No se permite la copia del vínculo y se muestra un mensaje de error indicando que se requiere un rol asociado para poder copiarlo.</p>	Vínculo sin un rol asociado en la plataforma.	Cumple	Se debe verificar que el sistema impida la copia de un vínculo que no tenga un rol asociado, evitando posibles inconsistencias y errores en la gestión de los autoregistros.
CP-HU-4791-005	Administrador	Verificar la eliminación de un autoregistro existente.	Existe al menos un autoregistro previamente	1. Iniciar sesión como administrador.	El autoregistro seleccionado se elimina correctamente de la	Autoregistro previamente creado en	Cumple	Se debe verificar que al eliminar un autoregistro, se eliminen de manera

ID	Rol	Descripción de la prueba	Precondición	Pasos	Resultado esperado	Set de datos	Ciclo	Observaciones
			creado en la plataforma.	2. Acceder a la sección de Gestión de autoregistros. 3. Seleccionar un autoregistro existente para eliminar. 4. Confirmar la acción de eliminar el autoregistro.	plataforma y se actualiza la tabla de autoregistros sin mostrar el registro eliminado.	la platafor ma.		adecuada todos los datos asociados a él y se actualice la interfaz de usuario para reflejar la eliminación. Además, se debe garantizar que no se produzcan errores o inconsistencias en la plataforma al realizar esta acción.

**Tabla 13.***Pruebas para Crud de Autoregistros - Lista de chequeo*

<b>Sección</b>	<b>Descripción de la prueba</b>	<b>Resultado esperado</b>	<b>Aplica S/N</b>	<b>Ciclo</b>	<b>Observaciones</b>
Tabla de registros	Validar que las columnas definidas en la HU coincidan con las Columnas de la funcionalidad	N/A	S	Cumple	No se registran observaciones
Tabla de registros	Validar que la tabla de registros tenga máximo 5 columnas y que los espacios entre las columnas se ajusten a la información y espacio definido.	N/A	S	Cumple	No se registran observaciones
Tabla de registros	Validar que los nombres de los campos se encuentren en modo oración	N/A	S	Cumple	No se registran observaciones
Tabla de registros	Validar que todos los botones de acción tengan su respectivo Tooltip	N/A	S	Cumple	No se registran observaciones
Tabla de registros	Validar que a medida que se van ingresando registros, el último en registrar quede en la primera fila del listado de registros.	N/A	S	Cumple	El orden de los autoregistros se organiza mediante la fecha desde menor.

<b>Sección</b>	<b>Descripción de la prueba</b>	<b>Resultado esperado</b>	<b>Aplica S/N</b>	<b>Ciclo</b>	<b>Observaciones</b>
Tabla de registros	Validar que la tabla de registros permita mostrar registros de 5, 10, 25 y 100 registros por página	N/A	S	Cumple	No se registran observaciones.
Tabla de registros	Validar que la tabla de registros permita pasar a la siguiente página	N/A	S	Cumple	No se registran observaciones
Tabla de registros	Validar que la tabla de registros permita regresar a la página anterior	N/A	S	Cumple	No se registran observaciones
Tabla de registros	Validar que al tener un número importante de registros el sistema permita ordenar la información de todas las columnas (Mayor a menor- Menor a Mayor - Alfabético. etc. (No aplica para la columna Acciones)	N/A	S	Cumple	No se registran observaciones
Tabla de registros	Validar que la opción Filtrar registros sólo se habilite cuando se hayan ingresado más de 5 registros	N/A	S	Cumple	No se registran observaciones
Tabla de registros	Validar que el sistema permite consultar información en la opción Filtrar registros y que muestre la respectiva información en la tabla	N/A	S	Cumple	No se registran observaciones

<b>Sección</b>	<b>Descripción de la prueba</b>	<b>Resultado esperado</b>	<b>Aplica S/N</b>	<b>Ciclo</b>	<b>Observaciones</b>
Tabla de registros	Validar que si hay más de tres botones de acción estos deben verse reflejados con los tres puntos (Ver estándares)	N/A	S	Cumple	No se registran observaciones
Tabla de registros	Validar que los valores numéricos en las tablas se encuentren discriminados a la derecha	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que el modal relacionado en la HU coincida con los campos relacionados en pantalla	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que los nombres de los campos se encuentren en modo oración	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que todos los botones de acción (Sólo si aplican dentro de un formulario tengan su respectivo Tooltip)	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que todos los campos del formulario tengan el icono en el costado izquierdo, excepto para el icono del calendario.	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que en los campos de listas desplegables y registro de información tengan sus respectivos textos indicativos.	N/A	S	Cumple	No se registran observaciones

Sección	Descripción de la prueba	Resultado esperado	Aplica S/N	Ciclo	Observaciones
Formularios	Validar que al dejar vacios campos obligatorios del formulario el sistema no permita guardar información y muestre mensaje de error - modo creación y edición	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que el sistema no permita ingresar registros duplicados (Debe especificarse en la HU) Ejemplo: Crear un permiso con los mismos datos de un registro ya creado)	N/A	S	Cumple	No se registran observaciones
Formularios	"Validar que al ejecutar la opción de Limpiar formulario el sistema elimine la información ingresada.  1. Para el caso del radio button estos deben aparecer con la misma selección que tenían antes de la modificación.  2. Para el caso de las fechas donde por defecto está marcada la fecha actual, esta debe borrarse al limpiar formulario."	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que la pantalla de consulta cumpla con los datos definidos en la HU y los formatos para datos estándar esten definidos correctamente	N/A	S	Cumple	No se registran observaciones

Sección	Descripción de la prueba	Resultado esperado	Aplica S/N	Ciclo	Observaciones
Formularios	Validar que todos los botones de acción tengan su respectivo Tooltip	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que el botón de cancelar y la (X) del modal, cierren la pantalla correspondiente.	N/A	S	Cumple	No se registran observaciones
Formularios	Validar que, al actualizar un registro, este se mantenga en la misma posición en la que se encontraba antes de su edición en el listado de registros.	N/A	S	Cumple	No se registran observaciones
Responsive - Dispositivo móviles	"Validar que los modales y menús correspondientes se adecuen a dispositivos móviles:  - Alineación  - Botones  - Listados  Tener en cuenta la dimensión mínima 360 "	N/A	S	Cumple	No se registran observaciones

## 9. Conclusiones

La modernización de la arquitectura ha sido un paso fundamental en la renovación de los sistemas de información de la Universidad Industrial de Santander. Mediante la sustitución de tecnologías obsoletas por herramientas y *frameworks* modernos, se ha logrado una mejora significativa en la eficiencia y la interacción del sistema, sentando las bases para futuras actualizaciones y mejoras, y asegurando su adaptabilidad a medida que avanza la tecnología.

En este contexto, las tareas desarrolladas en el sistema de información administrativa han desempeñado un papel crucial en la mejora del funcionamiento y en la entrega de una experiencia de usuario mejorada.

La implementación de la *HOME* de configuración general ha permitido centralizar y agilizar la administración y personalización del sistema, facilitando su manejo y adaptación a las necesidades específicas de cada usuario. Asimismo, la acción de generar nueva contraseña ha fortalecido la seguridad de los usuarios, brindando un método confiable para restablecer contraseñas y proteger la información sensible.

La zonificación del menú lateral izquierdo ha mejorado la navegación y la intuición del usuario al proporcionar una estructura clara y organizada, facilitando la localización de las opciones y mejorando la eficiencia en la ejecución de tareas. Además, la inclusión de un buscador específico para el menú de tercer nivel ha agilizado la navegación y el acceso a funcionalidades específicas, ahorrando tiempo y esfuerzo al usuario.

La implementación del CRUD de autoregistros ha permitido una gestión eficiente de los datos personales, brindando flexibilidad y control al usuario en la actualización y modificación de

su información. Esto ha contribuido a una experiencia más personalizada y se asegura la integridad de los datos.

La estabilización de los test de *frontend* ha sido un aspecto crítico para asegurar la calidad y confiabilidad del sistema, detectando posibles errores o fallas en la interfaz y corrigiéndolos de manera oportuna. Esto ha permitido ofrecer una experiencia consistente y libre de problemas a los usuarios.

Por último, la acción de pintar menús de acceso directo en la barra superior derecha ha mejorado la personalización, al brindar un acceso rápido a las opciones más utilizadas, lo que agiliza el acceso a funcionalidades clave y facilita la navegación en el sistema.

En resumen, estas tareas han contribuido significativamente a la optimización de los procesos y a una mayor satisfacción de los usuarios del sistema de información administrativa. Han mejorado la usabilidad, la seguridad y la eficiencia del sistema, permitiendo una gestión más efectiva de la información y de las funcionalidades. Además, han sentado las bases para futuras actualizaciones y mejoras, asegurando que el sistema de información administrativa se mantenga moderno, actualizado y acorde a las necesidades de la institución. La adopción del marco ágil SCRUM ha sido fundamental en este proceso, permitiendo una gestión eficiente del proyecto, la entrega incremental de funcionalidades y una mayor adaptabilidad a los cambios, lo que ha contribuido al éxito de la renovación de los sistemas de información.

## 10. Trabajo a Futuro

En un mundo en constante evolución tecnológica, es crucial mantener los sistemas de información actualizados y alineados con las últimas tendencias y avances. La renovación de los sistemas de información no es un proyecto de una sola vez, sino un compromiso continuo para garantizar la eficiencia, la seguridad y la adaptabilidad a medida que las necesidades cambian y las tecnologías avanzan.

Un trabajo a futuro de la renovación de los sistemas de información se basa en dos pilares fundamentales: las actualizaciones constantes de las herramientas utilizadas y el buen manejo del código y la arquitectura de software.

En cuanto a las herramientas utilizadas, como Angular, Spring Boot y Oracle Database, se debe tener en cuenta que estas tecnologías están en constante evolución. Es necesario seguir de cerca las actualizaciones y nuevas versiones para aprovechar las mejoras, las funcionalidades adicionales y los parches de seguridad. Mantenerse al día con estas actualizaciones garantizará un rendimiento óptimo, una mejor experiencia del usuario y la compatibilidad con las últimas tendencias y estándares del mercado.

Además, el buen manejo del código y la arquitectura de software juegan un papel crucial en la renovación de los sistemas de información. La adopción de buenas prácticas de desarrollo, como la modularización del código, el uso de patrones de diseño y la implementación de microservicios, permitirá una mayor flexibilidad y escalabilidad del sistema. Los microservicios, en particular, ofrecen la capacidad de descomponer las funcionalidades en componentes independientes, lo que facilita el mantenimiento, la actualización y la integración de nuevas características.

El trabajo a futuro implica también la adopción de marcos ágiles y prácticas de DevOps, que fomentan la colaboración, la entrega continua y la automatización de procesos. Estas metodologías y prácticas permiten una mayor eficiencia en el desarrollo, la detección temprana de errores y la reducción de tiempos de entrega.

En conclusión, esta visión nos permitirá mantener sistemas eficientes, seguros y adaptados a las necesidades cambiantes de las organizaciones, brindando una base sólida para la innovación y el crecimiento continuo.

### Referencias

- \_ESIC Business & Marketing School. (s. f.). Modelo entidad relación: descripción y aplicaciones. [www.esic.edu](https://www.esic.edu). <https://www.esic.edu/rethink/tecnologia/modelo-entidad-relacion-descripcion-aplicaciones>
- ¿Qué es .Net? Explicación de .Net - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/net/>
- Alex. (2021, 10 diciembre). GitKraken - Alex - Medium. Medium. <https://alexmarket.medium.com/gitkraken-ea27eb8e8301#:~:text=GitKraken%20es%20una%20herramienta,de%20proyectos%20entre%20otras%20cosas>.
- Angular. (s. f.). Desarrollo Web. <https://desarrolloweb.com/home/angular>
- API REST. (2021, 28 agosto). <https://www.ibm.com/co-es/cloud/learn/rest-apis>
- Arquitectura de Software. (s. f.). SG Buzz. <https://sg.com.mx/revista/27/arquitectura-software>
- Atlassian. (s. f.). Software de control de versiones: descripción general. Bitbucket. <https://bitbucket.org/product/es/version-control-software>
- Atlassian. (s. f.-a). Scrum: qué es, cómo funciona y por qué es excelente. <https://www.atlassian.com/es/agile/scrum>
- Carranza, A. (2021) ¿Qué es java y para qué sirve? Guía [2022]. <https://www.crehana.com/blog/transformacion-digital/que-es-java/>
- Casos de uso. (2017b, septiembre 15). Ingeniería de Software. <https://ingenieriadesoftwareanalis2017.wordpress.com/casos-de-uso/>
- Contributor, T. (2020, 30 octubre). GitLab. WhatIs.com. <https://www.techtarget.com/whatis/definition/GitLab>

D, D. (2022, 20 abril). Arquitectura de microservicios: qué es, ventajas y desventajas. Decide.

<https://decidesoluciones.es/arquitectura-de-microservicios/>

DevOps: qué es y cómo lo aplicamos. (2020, 21 julio). Claranet. [https://www.claranet.es/devops-](https://www.claranet.es/devops-que-es-y-como-lo-aplicamos)

[que-es-y-como-lo-aplicamos](https://www.claranet.es/devops-que-es-y-como-lo-aplicamos)

Education, I. C. (2021, 9 abril). Java Spring Boot. [https://www.ibm.com/cloud/learn/java-spring-](https://www.ibm.com/cloud/learn/java-spring-boot)

[boot](https://www.ibm.com/cloud/learn/java-spring-boot)

Hurtado, J. S. (2021, 7 diciembre). Metodología Scrum: qué es y cómo utilizarla para acometer

proyectos. Thinking for Innovation. [https://www.iebschool.com/blog/metodologia-scrum-](https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/)

[agile-scrum/](https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/)

Informix - Visión general. (s. f.-b). México | IBM. <https://www.ibm.com/mx-es/products/informix>

Introducción a JSON Web Token (JWT). (2020, 5 octubre). IONOS Digital Guide.

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/json-web-token-jwt/>

Jasmine Documentation. (s. f.). <https://jasmine.github.io/>

Las 5 ceremonias Scrum: claves para la gestión de procesos. (2019, 14 enero). Deloitte Spain.

<https://www2.deloitte.com/es/es/pages/technology/articles/ceremonias-scrum.html>

Morgan, A. (2020). Testing Angular with Jasmine and Karma (Part 1). DigitalOcean.

[https://www.digitalocean.com/community/tutorials/testing-angular-with-jasmine-and-](https://www.digitalocean.com/community/tutorials/testing-angular-with-jasmine-and-karma-part-1)

[karma-part-1](https://www.digitalocean.com/community/tutorials/testing-angular-with-jasmine-and-karma-part-1)

Qué es SCRUM. (2021, 20 septiembre). Proyectos Ágiles. [https://proyectosagiles.org/que-es-](https://proyectosagiles.org/que-es-scrum/)

[scrum/](https://proyectosagiles.org/que-es-scrum/)

Qué son Microservicios y ejemplos reales de uso. (2020, 4 junio). OpenWebinars.net.

<https://openwebinars.net/blog/microservicios-que-son/>

Siriwardhana, S. (2022). La Fácil Guía de Los Diagramas de Actividad UML. Blog de Creately.

<https://creately.com/blog/es/diagramas/tutorial-del-diagrama-de-actividad/>

Sistema de Información - Concepto, tipos, elementos y ejemplos. (s. f.). Concepto.

<https://concepto.de/sistema-de-informacion/>

Solutions, V., & Jain, A. (2023). Qué son los requisitos funcionales: ejemplos, definición, guía

completa. Visure Solutions. <https://visuresolutions.com/es/blog/requerimientos-funcionales/#:~:text=Los%20requisitos%20funcionales%2C%20como%20su,satisfacer%20las%20necesidades%20del%20usuario.>

Sotomayor, S. G. (2022, 8 marzo). Qué son las metodologías ágiles y cuáles son sus ventajas

empresariales. Thinking for Innovation. <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>

Technologies, U. (s. f.). CI/CD. Unity. <https://unity.com/es/solutions/what-is-ci-cd>

Techopedia. (2021, 21 junio). Oracle Database (Oracle DB). Techopedia.com.

<https://www.techopedia.com/definition/8711/oracle-database>

Toro, L. (2015, 4 diciembre). Taiga, the Best Agile Project Management Tool + Case Study. Desde

Linux. <https://blog.desdelinux.net/en/66643-2/>

Types of Actor in a Use Case Model. (s.f.). [https://www.visual-paradigm.com/guide/uml-unified-](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/types-of-actor-in-use-case-model/)

[modeling-language/types-of-actor-in-use-case-model/](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/types-of-actor-in-use-case-model/)