

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

MÓDULO DE CONFIGURACIÓN PARA SISTEMA DE GESTIÓN DE ALIMENTACIÓN XABAL SIGA VERSIÓN 2.0

Autores:

ALEJANDRO JAVIER MOLINA SANTOS
ENRIQUE FERREIRO ÁVILA

Tutores:

ING. YOSBEL FALERO VENTO
M.Sc. YADIRA RAMÍREZ RODRÍGUEZ

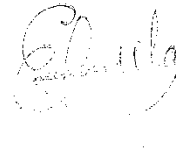
DECLARACIÓN DE AUTORÍA

Los autores del trabajo de diploma con título “**Módulo de Configuración para Sistema de Gestión de Alimentación XABAL SIGA Versión 2.0**” conceden a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declaran como únicos autores de su contenido. Para que así conste firma la presente a los 7 días del mes de Noviembre del año 2023.



Alejandro Javier Molina Santos

Autor



Enrique Ferreiro Ávila

Autor



M.Sc. Yadira Ramírez Rodríguez

Tutor



Ing. Yosbel Falero Vento

Tutor

DATOS DE CONTACTO

Yadira Ramírez Rodríguez, graduada de Ingeniería en Ciencias Informáticas en el año 2007. Máster en Calidad de Software y profesora auxiliar con más de 15 de años de experiencia en la asignatura de Ingeniería de Software. Jefa del Departamento Docente de Informática de la Facultad 4 de la Universidad de las Ciencias Informáticas. Líneas de Investigación: Ingeniería de Sistemas y Calidad de Software.

Correo electrónico: yramirezr@uci.cu

Yosbel Falero Vento, graduado de Ingeniería en Ciencias Informáticas en el año 2007, director del Departamento de Informatización de la Universidad de Ciencias Informáticas con más de 15 años de experiencia en el sector informático. Líder del proyecto Picta.

Correo electrónico: yfalero@uci.cu

AGRADECIMIENTOS

Alejandro Javier Molina Santos:

A Dios por permitirme cumplir mi sueño.

A mis padres por su sacrificio, amor inquebrantable y comprensión durante toda mi vida y en especial en mi etapa universitaria. Todos mis logros son para ustedes que han sabido ser ejemplo en todo para mí.

A mi hermana Ana, mi cuñado Rey Carlos y el más reciente miembro de la familia Thiago, gracias por estar siempre a mi lado.

A mis queridos abuelos, Damían, Isa, Zeida y Zeidita, por enseñarme a ser hoy un hombre de bien y darme un amor incomparable.

A Miriam y Tarrío por su preocupación constante y ayuda. Gracias por hacer desaparecer la palabra distancia.

A mi esposa Daniela por ser mi fiel compañera todos estos años y por su amor incondicional. Gracias por sostenerme y apoyarme en cada proyecto, este logro también es tuyo.

A mi tío Damían, mi tía Lourdes, y mis primas Daniela y Laia por brindar siempre su cariño a lo largo de los años y tenerme siempre presente.

A mi familia de Cienfuegos por permanecer cerca de mí y tratarme siempre con cariño desde que tengo uso de razón.

A mis suegros Dalay y Javier y a toda la familia de mi esposa por aceptarme, amarme, ser ejemplo de superación y de no rendirse ante las adversidades.

A Celín por su cariño y ayuda todos estos años, por enseñarme y aconsejarme.

A mi familia de Rovimpex, Abascal, Rene, Rabelo, Pedrini y Renata por estar a mi lado desde que nací, por ser esa mano amiga que nunca me ha abandonado.

A mis amigos, compañeros de carrera y profesores que a lo largo de los años me han aceptado y querido, sin ustedes no hubiese sido posible. En especial a Osmaní, Brian, Tony, el Pollo, Kevin, Castro y Yení, gracias de corazón por todo lo que me enseñaron.

A mis tutores por su esfuerzo y sacrificio para ayudarme en esta meta.

A mi compañero de tesis Enrique por ayudarme a lograr este objetivo común.

Enrique Ferreiro Ávila:

A mis padres, que siempre me apoyaron en cada paso de mi camino.

A mi tío Oscarito, mi tía Mayí, Erne y mis primos Oscar y Yuni, que siempre me ayudaron en todo lo que necesité.

A mi hermano, que siempre creyó en mí y se preocupó por todo.

A mis abuelos Isidoro, Alma y Goyo que en paz descanse, espero estén orgullosos de mí.

A Aarom, que me ayudó siempre que pudo y me brindó sus conocimientos. Gracias por compartir tu amistad conmigo.

A Beatriz y Milena que estuvieron ahí ayudándome en lo que pudieron haciéndome el camino más fácil y divertido.

A Carlos y Roxana que también me ayudaron siempre que se pudo, les agradezco su apoyo en cada momento.

A Daniela y Yanitza que también estuvieron presentes en todo .Gracias por su amistad y apoyo en este proceso largo.

Al resto de mis amigos en especial para Richard, Roberto, José Carlos, Dayanís, Nathali, Rosmery y Kevin, que también aportaron de una manera u otra a mi graduación.

A mi compañero de tesis Molina, que me ayudó en todo formando un gran equipo y logrando un trabajo del que estamos orgullosos.

A mis tutores, que me guiaron para lograr hacer una tesis de calidad y graduarme. Les estoy muy agradecido por su paciencia, orientación y consejos.

.

DEDICATORIA

Alejandro Javier Molina Santos:

A mí abuelo Pitín, por su entrega y amor infinito.

Espero que estés orgulloso desde el cielo.

Enrique Ferreiro Ávila:

A mis padres, por darme la vida, la educación y el ejemplo a seguir. Gracias por creer en mí y por acompañarme en cada momento, sin ustedes nada de esto hubiera sido posible. Esta tesis es el fruto de su esfuerzo y su dedicación. Los quiero mucho.

RESUMEN

La Dirección de Informatización de la Universidad de las Ciencias Informáticas en Cuba apuesta por la actualización del Sistema de Gestión de Alimentación a una versión 2.0 con el fin de brindar una aplicación que permita la modificación de sus componentes. Esto permitirá que se adapte a necesidades de clientes diferentes con vistas a una futura comercialización del mismo. Tomando como referencia la importancia de contar con una configuración flexible que permita la modificación de parámetros para ajustarse a las necesidades del usuario final. En la presente investigación se plantea como objetivo principal desarrollar un módulo de configuración para el Sistema de Gestión de Alimentación XABAL SIGA Versión 2.0 que cumpla con las expectativas de la Dirección de Informatización de la Universidad. En el desarrollo de este componente se utiliza un entorno de desarrollo integral que comprende Python con Django REST framework, JavaScript con React y PostgreSQL como base de datos principal. Para el cumplimiento de este objetivo se definen los fundamentos teóricos, se realiza un análisis y diseño de la propuesta de solución, hasta la aplicación de pruebas unitarias y de aceptación con vistas a comprobar el correcto funcionamiento de la solución propuesta, empleando como guía fundamental la metodología ágil Programación Extrema, por sus siglas en inglés XP.

PALABRAS CLAVES

Módulo de configuración, XABAL SIGA, Sistema de Gestión de Alimentación.

ABSTRACT

The Directorate of Informatization at the University of Computer Sciences in Cuba is committed to updating the Food Management System to version 2.0 in order to provide an application that allows the modification of its components. This will enable it to adapt to the needs of different clients, with a view to potential commercialization. Taking into consideration the importance of having a flexible configuration that allows parameter modifications to align with the requirements of the end user, this research aims to develop a configuration module for the Food Management System XABAL SIGA Version 2.0 that meets the expectations of the University's Directorate of Informatization. In the development of this component, an integrated development environment is used, including Python with Django REST framework, JavaScript with React, and PostgreSQL as the primary database. To achieve this objective, theoretical foundations are defined, a solution proposal is analyzed and designed, and unit and acceptance tests are performed to verify the proper functioning of the proposed solution, following the fundamental guidance of the agile methodology Extreme Programming (XP).

KEYWORDS

Configuration module, XABAL SIGA, Feed Management System.

ÍNDICE

INTRODUCCIÓN.....	- 1 -
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA	- 5 -
1.1 Configuración en sistemas informáticos	- 5 -
1.2 Sistemas de gestión de alimentación	- 7 -
1.2.1 Sistema de Gestión de Alimentación XABAL SIGA.....	- 9 -
1.3 Diagnóstico del Módulo de Configuración del Sistema XABAL	- 10 -
SIGA.....	- 10 -
1.4 Análisis del módulo de configuración de soluciones similares existentes.....	- 13 -
1.4.1 Sistema Integrado de Gestión Administrativa del Ministerio de Economía y Finanzas SIGA – MEF.....	- 13 -
1.4.2 Sistema de Gestión Administrativa Escolar SIGAEWEB.....	- 15 -
1.4.3 Distra	- 16 -
1.4.4 Análisis del estudio realizado a los módulos de configuración de soluciones existentes.	- 18 -
1.5 Estudio de herramientas y tecnologías.....	- 20 -
1.5.1 Lenguajes de programación para el desarrollo del lado del cliente.....	- 20 -
1.5.2 Frameworks de desarrollo del lado del cliente.	- 21 -
1.5.3 Otras tecnologías utilizadas en el desarrollo del lado del cliente.	- 23 -
1.5.4 Lenguajes de programación para el desarrollo del lado del servidor.	- 24 -
1.5.5 Frameworks de desarrollo web para el lado del servidor.	- 28 -
1.5.6 Sistema gestor de bases de datos	- 30 -
1.5.7 Otras herramientas.....	- 31 -
1.6 Metodología.....	- 32 -
1.7 Conclusiones del capítulo.....	- 35 -
CAPÍTULO II: ANÁLISIS Y DISEÑO DEL SISTEMA	- 36 -
2.1 Propuesta de solución.	- 36 -
2.2 Usuarios	- 39 -

2.3 Fase 1: Planificación.....	- 39 -
2.3.1 Historias de Usuario (HU).....	- 40 -
2.3.2 Requisitos no funcionales.	- 44 -
2.3.3 Estimación de esfuerzos por HU	- 45 -
2.3.4 Plan de iteraciones.....	- 47 -
2.3.5 Plan de entregas	- 49 -
2.4 Fase II: Diseño del sistema.	- 50 -
2.4.1 Diseño arquitectónico.....	- 50 -
2.4.2 Patrones de diseño	- 52 -
2.4.3 Tarjetas Clase - Responsabilidad – Colaborador (CRC).....	- 54 -
2.8 Conclusiones del capítulo.....	- 55 -
CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA.....	56
3.1 Fase III: Desarrollo	56
3.1.1 Tareas de ingeniería	56
3.2 Fase IV: Pruebas.....	58
3.2.1 Pruebas unitarias.	59
3.2.2 Pruebas de aceptación.....	62
3.3 Conclusiones del capítulo.....	65
CONCLUSIONES FINALES	- 66 -
RECOMENDACIONES.....	- 67 -
REFERENCIAS BIBLIOGRÁFICAS	- 68 -
ANEXOS.....	- 77 -

ÍNDICE DE TABLAS

Tabla 1. Características generales de las soluciones estudiadas.....	- 18 -
Tabla 2. Características generales de las tecnologías estudiadas. .	- 22 -
Tabla 3. Comparación entre lenguajes de programación para desarrollo del lado del servidor.	- 27 -
Tabla 4. Descripción por usuario del sistema.	- 39 -
Tabla 5. Historia de usuario 1 Gestionar clasificación de platos.	- 41 -
Tabla 6. Historia de usuario Gestionar unidades de medida..	- 42 -
Tabla 7. Estimación de esfuerzos por HU.....	- 45 -
Tabla 8. Plan de iteraciones.	- 48 -
Tabla 9. Fechas de entregables.	- 49 -
Tabla 10. Tarjeta CRC Clasificación de platos	- 54 -
Tabla 11. Tarjeta CRC Unidades de medida.	- 55 -
Tabla 12. Tarea de ingeniería No.1 de la HU Gestionar clasificación de platos.	56
Tabla 13. Tarea de ingeniería No.2 de la HU Gestionar clasificación de platos.:	57
Tabla 14. Prueba de aceptación del Gestionar clasificación de platos.	62
Tabla 15. Prueba de aceptación del Gestionar unidades de medida.....	64

ÍNDICE DE FIGURAS

Figura 1. Componentes del módulo de configuración actual del sistema XABAL SIGA	- 11 -
Figura 2. Propuesta de solución..	- 37 -
Figura 3. Diseño arquitectónico para XABAL SIGA Versión 2.0..	- 51 -
Figura 4. Pruebas unitarias al Gestionar clasificación de platos.....	60
Figura 5. Resultados de pruebas unitarias realizadas al Gestionar clasificación de platos. ...	61
Figura 6. Comparativa de resultados de pruebas unitarias.....	61
Figura 7. Resultados finales de pruebas unitarias realizadas.....	62

INTRODUCCIÓN

Desde hace varias décadas, la informatización de los procesos ha sido una herramienta fundamental para el desarrollo de la humanidad. Esta permite captar, procesar, tener acceso a información que antes no estaba disponible y generar nuevos conocimientos, con una eficiencia superior (Baluja & Soler, 2020). En las últimas décadas, el vertiginoso progreso en el campo de la informática ha dado paso a sistemas y aplicaciones cada vez más versátiles y personalizables (Cubides, 2020).

En la era actual, la personalización y configuración de software se ha convertido en un aspecto crucial para satisfacer las necesidades específicas de los usuarios y las empresas (Gutierrez D. , 2020). Los avances en el diseño de software, la arquitectura de sistemas y la programación orientada a objetos han permitido la creación de módulos y componentes que pueden ser configurados de manera independiente. Esto ha llevado a la creación de aplicaciones altamente modulares y adaptables.

La necesidad de configurar se ha acentuado con la diversidad de usuarios y contextos de uso. Las organizaciones requieren soluciones que se ajusten a sus flujos de trabajo específicos, mientras que los usuarios individuales buscan una experiencia que se adapte a sus preferencias personales. Esto ha sido una necesidad para todo tipo de aplicaciones y en todo tipo de sectores de la sociedad, incluyendo en las dirigidas a la gestión de los procesos de alimentación.

Los sistemas de gestión de alimentación directamente brindan una mejoría desde la precisión de datos estadísticos, el control de personal y la planificación de comensales (Javaria Akram , 2021; Kim, 2020). El uso de estos sistemas se ha vuelto cada vez más común en los últimos años gracias a los avances tecnológicos y la necesidad de mejorar la calidad de los servicios de alimentación en las instituciones.

En Cuba, en el año 2002 se funda la Universidad de las Ciencias Informáticas (UCI), ciudad tecnológica avanzada que cuenta con acreditación y reconocimiento a nivel internacional. Este centro forma profesionales en la rama de la informática, siendo un pilar para el proceso de informatización de la isla (Universidad de las Ciencias Informáticas, 2023). Esta cuenta

con un sistema dirigido a la gestión del servicio de alimentación de los comensales de la institución XABAL SIGA, por sus siglas Sistema Integrado de Gestión de Alimentación.

XABAL SIGA fue implementado con una arquitectura Modelo-Vista-Controlador, lo cual dificulta la personalización del mismo en cualquier otro centro que requiera este tipo de solución informática, implicando desplegar todos sus módulos aunque no fuesen necesarios para el centro en cuestión. Los componentes de este sistema presentan interdependencia, no permiten la desagregación de los mismos con facilidad, esto debido en gran manera a la arquitectura antes mencionada (Fowler, 2002). Todo esto puede aumentar la complejidad y el riesgo de producirse errores en el sistema si un cliente necesita hacer cambios específicos en un componente. Con ello, se requiere un mayor esfuerzo desde el punto de vista de implementación y por tanto aumenta el tiempo para dar respuesta a la solicitud del cliente.

Por otra parte, presenta una importante deuda tecnológica, pues fue desarrollado con PHP 7.2 y su *framework* Symfony 2.8, que ya no reciben actualizaciones de seguridad y corrección de errores, presentando vulnerabilidad ante virus y ciberataques. Por último, este escenario claramente limita la comercialización de este producto por lo que se restringe la oportunidad de generar ingresos a la universidad.

Considerando estos elementos, esta investigación está dirigida a solucionar el siguiente problema de investigación: ¿Cómo puede convertirse el XABAL SIGA Versión 2.0 en un producto personalizable según las necesidades de cada cliente?

Para dar solución al problema de investigación se propone como objetivo general desarrollar un módulo de configuración para el Sistema de Gestión de Alimentación XABAL SIGA Versión 2.0. Se define como objeto de estudio: los módulos de configuración de sistemas informáticos. El campo de acción: el módulo de configuración para el Sistema de Gestión de Alimentación de la Universidad de las Ciencias Informáticas.

Con vistas a dar cumplimiento al objetivo planteado, se proponen las siguientes tareas de la investigación:

- Elaboración del marco teórico de la investigación a través de un estudio del estado del arte para facilitar la comprensión de la investigación.

- Realización del levantamiento de requisitos funcionales y no funcionales del módulo a desarrollar para definir qué debe hacer la aplicación y qué características debe tener.
- Elaboración de la planificación y diseño de la solución propuesta para satisfacer tanto los requisitos del cliente como los plazos de entrega.
- Desarrollo del módulo de configuración con el propósito de cumplir el objetivo general de esta investigación y resolver la problemática planteada.
- Realización de pruebas de aceptación con el fin de comprobar el correcto funcionamiento de los requisitos implementados.

Durante el proceso de investigación se emplearon métodos científicos los cuales se exponen a continuación:

Métodos teóricos:

- **Modelación:** Se modeló el desarrollo de la solución a partir de los artefactos que establece la metodología de desarrollo y el modelo de datos de la solución propuesta.
- **Análisis sintético:** Permitió concretar y resumir el conocimiento obtenido de los materiales estudiados sobre la gestión de alimentación y los sistemas de gestión de alimentación, así como la descripción y análisis de los procesos observados con el fin de emplearlo en el desarrollo de la presente investigación.

Métodos empíricos:

- **Observación:** Facilitó obtener conocimiento acerca del funcionamiento del sistema actual y los procesos de gestión de alimentación en la universidad.

La presente investigación está compuesta por tres capítulos de los cuales se brinda una breve descripción a continuación:

Capítulo 1 Fundamentación teórica: En esta primera etapa se aborda el estado del arte de la investigación, que incluye un estudio bibliográfico de los conceptos clave relacionados con

el tema y la realización de un diagnóstico del módulo de configuración del Sistema de Gestión de Alimentación. Durante este proceso se elige el entorno de desarrollo apropiado para la propuesta y se selecciona la metodología de desarrollo con el objetivo de garantizar la generación de los artefactos necesarios para la documentación del desarrollo.

Capítulo 2 Análisis y diseño del sistema: En este capítulo se presenta una propuesta de solución para los problemas detectados, partiendo del estudio realizado en la fundamentación teórica. Se mencionan los usuarios que intervienen en la solución. También se definen las Historias de Usuario, la planificación de entrega de versiones del producto y el diseño del sistema de acuerdo a las fases que propone XP.

Capítulo 3 Implementación y pruebas del sistema: En este capítulo se describen las iteraciones realizadas durante la etapa de desarrollo del módulo de configuración del Sistema de Gestión de Alimentación XABAL SIGA en su versión 2.0. Además se definen las tareas de ingenierías correspondientes a cada historia de usuario. También se aplican las pruebas unitarias y de aceptación que permiten comprobar el correcto desarrollo de las funcionalidades implementadas de acuerdo con lo propuesto por la metodología Programación Extrema.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Este capítulo se centra en proporcionar una base teórica sólida que respalde y justifique la investigación realizada. Estará dedicado a la búsqueda y análisis de la información relacionada con el dominio del objetivo general. Se comienza sintetizando algunas conceptualizaciones fundamentales relacionadas a los sistemas de gestión de alimentación y módulos de configuración. En este orden, serán analizadas algunas soluciones existentes referentes al mismo entorno. Para finalizar, se realiza la selección y descripción de las principales tecnologías, metodologías y herramientas que serán utilizadas en el desarrollo de la propuesta de solución.

1.1 Configuración en sistemas informáticos

La configuración de un sistema se refiere a los ajustes y parámetros que se pueden personalizar para modificar el comportamiento del mismo a las necesidades y preferencias del usuario (Randal E. Bryant, 2021). Estos abarcan una amplia gama de aspectos, desde opciones básicas hasta configuraciones más complejas y avanzadas. Las configuraciones generalmente se cargan durante el inicio del programa y en algunos casos es necesario reiniciarlo para poder aplicar cambios. Es habitual que los programas utilicen ficheros para guardar su configuración, pero también puede almacenarse en una base de datos. Aunque se pueden encontrar programas que cifran su configuración para evitar que los usuarios la modifiquen.

Todas las aplicaciones informáticas cuentan con una configuración predeterminada, la cual se refiere a los ajustes y parámetros preestablecidos de un sistema que se aplican cuando se instala o se inicia por primera vez sin ninguna personalización por parte del usuario (Larry L. Peterson, 2019). Esta configuración inicial está diseñada para proporcionar un funcionamiento básico y genérico del sistema. Sin ella, en la mayoría de los casos es imposible que se inicialice la aplicación. Según (Norman, 2013) y (Tidwell, 2010) los menús de ajustes en aplicaciones para gestión de elementos suelen tener algunas de las siguientes características generales:

- **Organización en categorías:** Los ajustes suelen organizarse en categorías o pestañas temáticas para facilitar la navegación y búsqueda de opciones específicas.
- **Submenús:** Dentro de cada categoría, es común encontrar submenús que agrupan ajustes relacionados, lo que ayuda a mantener una estructura jerárquica.
- **Iconos y descripciones:** Cada opción de configuración suele ir acompañada de un icono y una descripción breve para ayudar a los usuarios a comprender su propósito.
- **Valores ajustables:** Los menús de configuración permiten a los usuarios modificar valores y preferencias, como activar o desactivar funciones, registrar usuarios, seleccionar preferencias, entre otros...
- **Seguridad y autenticación:** Los ajustes sensibles, como las preferencias de seguridad o privacidad, a menudo requieren autenticación adicional para evitar cambios no autorizados.
- **Acceso basado en roles y permisos:** Suelen permitir la configuración de roles y permisos para usuarios, y esta configuración se realiza a través de los ajustes. Los administradores pueden asignar diferentes niveles de acceso y capacidades a los usuarios.
- **Configuración de usuarios y perfiles:** En las configuraciones de aplicaciones de gestión los administradores pueden utilizar el menú de configuración para gestionar usuarios y perfiles. Esto puede incluir la creación, modificación y eliminación de cuentas de usuario, así como la asignación de roles y permisos específicos a cada usuario. Además, se pueden establecer preferencias de usuario individuales, como la imagen de perfil, la dirección de correo electrónico, las notificaciones específicas y las preferencias de visualización, lo que permite una experiencia altamente personalizada para cada usuario.

Teniendo en cuenta estos elementos como base para el desarrollo del módulo de configuración, cabe destacar que la importancia de la configuración radica en la capacidad de perso-

nalizar el sistema según las necesidades y preferencias individuales. Permite a los usuarios adaptar la experiencia de uso, mejorar la eficiencia y la productividad, y ajustar la seguridad y el rendimiento según los requisitos específicos. (Thomas A. Limoncelli, 2020), (Robert Aiello, 2019). La personalización es una excelente manera de mejorar las experiencias de los usuarios, les permite decidir específicamente qué desean ver, lo que significa que solo mostrará el contenido que sus usuarios desean. (ZAPROO, 2022). La configuración precisa garantiza la consistencia y la coherencia en el entorno de *software*. Al establecer las opciones, se evitan discrepancias y comportamientos inesperados en el sistema. Esto es especialmente importante en entornos de desarrollo y producción, donde se requiere que los sistemas sean consistentes en diferentes instancias y entornos. También, establecer ajustes precisos ayuda a cumplir con los requisitos de seguridad, cumplimiento normativo y buenas prácticas establecidas.

Por otra parte, la gestión de la configuración se refiere a la práctica de administrar y controlar los elementos de configuración de un sistema a lo largo de su ciclo de vida. Involucra actividades y procesos para identificar, documentar, controlar y mantener de manera sistemática los elementos de configuración, como componentes de software, hardware, documentación, datos y configuraciones asociadas (Gene Kim, 2022). Esto es esencial para garantizar la integridad, la coherencia y la trazabilidad de los sistemas y software a lo largo de su ciclo de vida. Ayuda a evitar errores, mejorar la eficiencia en el desarrollo y mantenimiento, facilitar la colaboración entre equipos y brindar un mayor control sobre los cambios realizados en el software. Teniendo en cuenta estos elementos, es preciso el estudio de los Sistemas de Gestión de Alimentación con el fin de comprender el funcionamiento de este tipo de aplicaciones.

1.2 Sistemas de gestión de alimentación

Existen diversas perspectivas teóricas referentes al término de sistema de gestión de alimentación. En este contexto, es posible dividirlo en conceptos que permiten entender su funcionamiento. Según la definición del Diccionario de la Lengua Española un *sistema* “es un conjunto de elementos que interactúan entre sí para lograr un objetivo común” (Real Academia

Española, 2014). Esta conceptualización es importante pero básica, por lo que orientado más al entorno de las ciencias informáticas se tiene que, un sistema informático se refiere al conjunto de elementos que procesan, almacenan y transmiten información (Kenneth C. Laudon, 2012).

La utilización de software informático se ha extendido a todos los sectores en la sociedad, Pressman (2010) describió que en las instituciones, esto permite automatizar y optimizar los procesos de negocio, lo que a su vez contribuye a aumentar la eficiencia, la productividad y la toma de decisiones informadas. Todo lo anterior evidencia la importancia de contar con herramientas que mejoren los procesos en todos los ámbitos, por lo que no queda exento de ello el área de alimentación.

En el sector de la alimentación la informatización de actividades, como la gestión de órdenes de clientes y de inventarios, hacen de estas aplicaciones una herramienta indispensable. Teniendo en cuenta el enfoque de este tipo de aplicaciones en la industria, se puede definir sistema de gestión de alimentación (SGA).

SGA es el conjunto de procedimientos destinados a organizar, dirigir, administrar, evaluar y controlar actividades, recursos humanos y procesos del servicio de alimentación. Con el objetivo de brindar alimentación con los más altos estándares de calidad y garantizar que las necesidades de los comensales que demandan el servicio sean satisfechas. (Leal, 2022). Dentro de esto también es posible realizar un mejor control de comedores y cafeterías.

El control de comedores facilita la gestión de personal y alimentos, permite determinación de horarios distribución de personal, menú, porciones de alimentos, de esta manera se establecen cifras que indican cuánta comida debe servirse, evitando desperdicios. (Briseño, 2021). Además, este tipo de sistemas por lo general, cuentan con un módulo para establecer precios a los alimentos y costo por platillos, lo que es considerado una herramienta valiosa para instituciones que desean llevar una contabilidad rigurosa del proceso de alimentación. (Ingeaa, 2023).

Un sistema de gestión de alimentación por las características descritas puede ser implementado en universidades, escuelas, centros de trabajo, hospitales, entre muchas otras organi-

zaciones, otorgándole esto un amplio mercado. La Universidad de las Ciencias Informáticas cuenta con una solución informática de este estilo, el Sistema de Gestión de Alimentación XABAL SIGA.

1.2.1 Sistema de Gestión de Alimentación XABAL SIGA

El sistema de gestión de alimentación XABAL SIGA fue desarrollado por la Dirección de Informatización de la Universidad de las Ciencias Informáticas (UCI) en Cuba. Este brinda la posibilidad de realizar la asignación de comensales, atendiendo a una determinada estructura de comedores. Facilita gestionar distribuciones de comensales para fechas y períodos especiales, además de ofrecer reportes como cantidad de comensales que han recibido el servicio y desglosarlo por puerta o por tipo. Este sistema consta de múltiples módulos destinados al proceso de alimentación en una universidad, entender su funcionamiento es fundamental para el desarrollo de configuraciones que se ajusten a estos componentes. A continuación se describen cada uno de ellos:

- **Abastecimiento:** Permite visualizar a través de gráficos los productos en almacén y gestionar platos, menús y productos. Además facilita la planificación para fechas y períodos, lo cual se muestra a través de imágenes para una mejor comprensión del usuario.
- **Reservación:** Permite realizar una reserva para obtener acceso al servicio de alimentación de la universidad, estas reservas se realizan a través de una interfaz donde se muestra el menú para el día y pueden ser para personal de la universidad, familiares, asociados y terceros. Para ello también es posible visualizar un reporte de platos y menús por fechas.
- **Distribución:** Permite visualizar gráficas de reservaciones por días y tarjetas distribuidas. Este módulo facilita la organización de comensales por los centros de alimentación que define la universidad.
- **Cajero:** Permite visualizar de forma gráfica la cantidad de accesos por complejo de alimentos y eventos. Además a través de este módulo es posible el escaneo de

tarjetas antes de recibir el servicio, como forma de control del personal, visualizando en pantalla los datos de la persona que entra al comedor, el evento en el que se encuentra y la fecha y hora.

- **Facturación:** Muestra gráficamente por fecha la cantidad de dinero facturado por la universidad en cada evento: desayuno, almuerzo y comida. Además, es posible el control de esto a través de listados con los importes correspondientes a cada fecha.
- **Reportes:** Este módulo brinda datos estadísticos sobre todo el proceso de gestión de alimentación de la universidad. Dentro de este marco es posible visualizar un parte general a partir de la introducción de una fecha por el usuario y un parte específico por diferentes filtros.
- **Administración:** Permite al usuario realizar gestión de roles, permisos, usuarios y grupos de roles.

Por último se encuentra el módulo de “Configuración”, el cual establece los principios fundamentales para el funcionamiento del negocio y se detalla en el diagnóstico que se le realiza a continuación.

1.3 Diagnóstico del Módulo de Configuración del Sistema XABAL

SIGA.

Un módulo de configuración es un subsistema que puede ser conectado o desconectado de un sistema mayor para modificar las propiedades estructurales o funcionales del sistema total. Los módulos de configuración permiten crear sistemas adaptables donde se pueden cambiar partes específicas sin necesidad de modificar del todo la aplicación (Padilla, 2013). Este apartado es el regidor de todas las funcionalidades de la aplicación y de su personalización. Finalmente, debe integrarse con otros componentes del sistema para aplicar los cambios de configuración de manera adecuada. Esto puede implicar la comunicación con otros módulos, reinicio de servicios o actualización de parámetros en tiempo real, según sea necesario.

Como se muestra en la Figura 1. Componentes del módulo de configuración actual del sistema XABAL SIGA. los elementos que componen el módulo de configuración del sistema XABAL SIGA son:



Figura 1. Componentes del módulo de configuración actual del sistema XABAL SIGA. Fuente: Elaboración propia.

En su primera versión, XABAL SIGA ofrece configuraciones específicas relacionadas con el módulo de "Abastecimiento". Una de estas configuraciones se centra en la gestión de la clasificación de platos, permitiendo categorizar los alimentos en diferentes grupos como cereales, caldos, cremas, entre otros. Además, es posible ajustar las unidades de medida, lo que proporciona una funcionalidad para la conversión automática de las unidades. Asimismo, desde el apartado de "Abastecimiento", se puede gestionar los tipos de productos, establecer categorías y administrar los productos almacenados. Este, también brinda la posibilidad de planificar el menú.

En otro orden de elementos, en los ajustes del sistema, se puede modificar los parámetros relacionados con el módulo de "Cajero". Estos ajustes permiten al usuario gestionar las tarjetas de acceso de los comensales y asignar direcciones IP a las puertas utilizadas por los cajeros. En la sección de configuraciones del módulo de "Cajero", también se puede manejar la gestión de los torpedos (visitantes ocasionales a la universidad con acceso al servicio de alimentación) y la gestión de solapines perdidos. Además, se incluye la posibilidad de modificar

parámetros específicos para el módulo de "Facturación", donde se establecen reglas de excepción para recibir el servicio.

Dentro del módulo se encuentra "Distribución", donde almacena las configuraciones relacionadas con este sector del sistema. Aquí se gestionan las estructuras y los eventos (desayuno, almuerzo y cena) junto con sus horarios y rangos correspondientes. También brinda facilidades para la gestión del personal y su distribución dentro de la institución. Otra sección se dedica a las configuraciones de "Reservación", donde se establecen los períodos y los responsables de este proceso. Por último se encuentran los ajustes más generales de comensales, personas por evento y cobro. Todo esto acompañado de las tareas programadas y los avisos.

Otro elemento del módulo de configuración es la seguridad, donde se realizan funciones de gran importancia como la gestión de sistemas y de componentes. Además aspectos imprescindibles para el negocio como las agrupaciones funcionales, roles, grupos de roles y funcionalidades del sistema. Por último desde aquí es que se realiza la gestión de los usuarios donde se puede ver los permisos que posee cada uno.

Aunque el módulo actual del Sistema de Gestión de Alimentación permite una modificación del proceso de alimentación, no es efectivo en su totalidad. La principal deficiencia del mismo se encuentra en la imposibilidad de desagregación de la estructura del sistema, no permite que los componentes sean seleccionables e independientes, con el fin de ajustarse a necesidades diferentes a las de la universidad. Además, dentro de la sección de ajustes de "Abastecimiento" se gestiona la clasificación de las unidades de medida, lo cual es innecesario y provoca un aumento en la carga del sistema.

En adición a lo antes expuesto, para el funcionamiento correcto del sistema actualmente siempre debe existir una reservación activa, lo cuál debe ser más flexible. También el sistema está diseñado para una realidad diferente, ya que la facturación y costo de los productos está diseñado en las monedas CUC y CUP, y actualmente en el país solo circula el CUP.

En otro orden, el módulo de configuración del XABAL SIGA fue desarrollado en PHP 7.2, que ha llegado al final de su ciclo de vida útil en noviembre de 2020 y, por lo tanto, ya no recibe

actualizaciones de seguridad. También se utilizó como *framework* de desarrollo Symfony 2.8 el cual es una versión antigua de un *framework* PHP que ya no se encuentra en soporte activo. Esto significa que cualquier vulnerabilidad descubierta después de la fecha de finalización del soporte no será corregida por los desarrolladores y, por lo tanto, los problemas de seguridad permanecerán sin solución. Esto representa un riesgo potencial para la integridad, confidencialidad, disponibilidad de los datos y funcionalidades del sistema, por lo que se recomienda actualizar el módulo a tecnologías más recientes y soportadas. Con el fin de un desarrollo acertado para la nueva propuesta de solución se hace necesario examinar los módulos de configuración de sistemas similares existentes en el mercado.

1.4 Análisis del módulo de configuración de soluciones similares existentes.

Con el objetivo de identificar las principales características que debe poseer un módulo de configuración en sistemas de gestión, se realiza un estudio de este apartado en diferentes sistemas informáticos que existen actualmente en el mercado. Actualmente es imprescindible para la mayoría de soluciones informáticas contar con un producto configurable y adaptable. Es por esto que en los siguientes sub epígrafes se realizará un análisis de la configuración de plataformas que gestionen recursos de forma similar a XABAL SIGA tanto a nivel internacional como nacional.

1.4.1 Sistema Integrado de Gestión Administrativa del Ministerio de Economía y Finanzas SIGA – MEF.

El Sistema Integrado de Gestión Administrativa es una aplicación informática desarrollada por la Oficina General de Tecnología de la Información (OGTI) del Ministerio de Economía y Finanzas de Perú (Farro, 2016). Este proyecto se originó en el año 2000 como una necesidad, con la finalidad de lograr la integración de todos los sistemas administrativos en uno solo (R&C Consulting, 2019). Entre las herramientas que presenta contiene una sección “Patrimonio” donde se gestionan los bienes estatales de muebles e inmuebles. Otras herramientas del sistema son la gestión de la “Logística”, “Presupuesto”, “Bienes corrientes” y “Tesorería”. La importancia de este sistema se basa en la mejora de la eficiencia de la Gestión Pú-

blica de los Procesos de Abastecimiento y Control Patrimonial. Las ventajas de contar con el SIGA es que permite la mejora de calidad de ejecución de gastos, ahorro en recursos y ordenamiento (Bludid, 2019).

El módulo de configuración de este sistema comprende las funcionalidades de configuración de Logística, Sedes, Feriados, Centros de Costo, Personal, Proveedores, Configuración de la Programación y Carga de Datos del Siga (Ministerio de Economía y Finanzas, 2021). Este, contiene una serie de elementos de importancia dentro del sistema como son: guardar los datos ingresados en esa ventana, actualizar los datos, visualizar los reportes, imprimirlos y exportar estos reportes a Excel o PDF. La barra de menú ubicada en la parte superior de la ventana de configuración incluye un conjunto de componentes utilizados para operar los ajustes del sistema. Su descripción y funcionalidades se detallan a continuación:

Administración: Contiene las funcionalidades que permiten al Usuario cambiar su clave de acceso al Sistema. Asimismo, contiene la opción que permite cerrar la sesión del usuario, retornando a la ventana de Inicio de Sesión (R&C Consulting, 2019).

Maestros: Permite consultar y cargar ítems en Catálogo de Bienes y Servicios, personalizar el catálogo Institucional, así como Registrar los Datos Generales como Marcas, Sedes, Feriados, Tareas. Además, el registro de Centros de Costo, Personal, Proveedores, Ubicación Física y Colores que permitirán la operatividad de las funcionalidades en los diferentes Módulos (R&C Consulting, 2019).

El desarrollo del SIGA-MEF involucró el uso de diferentes lenguajes de programación, dependiendo de las necesidades y requerimientos de cada módulo o componente del sistema. Los lenguajes utilizados según (Luicho, 2020) en el desarrollo del SIGA-MEF son:

- **C#:** es un lenguaje de programación orientado a objetos que fue utilizado para desarrollar varios componentes del sistema, incluyendo la capa de servicios y la capa de acceso a datos.
- **PL/SQL:** es un lenguaje de programación utilizado para desarrollar procedimientos almacenados y funciones en la base de datos Oracle, que es la base de datos utilizada por el SIGA-MEF.

- JavaScript: es un lenguaje de programación utilizado principalmente para desarrollar la lógica del lado del cliente en las páginas web del sistema.
- *HyperText Markup Language* (HTML) / *Cascading Style Sheets* (CSS): son lenguajes de marcado utilizados para desarrollar la estructura y el estilo de las páginas web del sistema.

Por último es importante destacar que es una aplicación de escritorio que no presenta posibilidades de modificar su estructura o contenido (Farro, 2016). Por tanto, no es posible disminuir la carga de la aplicación retirando alguna sección en caso de no ser necesaria o utilizada. Entre las principales deficiencias actuales del sistema está el no contar con un seguimiento de algunos de los módulos desarrollados en sus primeras versiones. Teniendo esto en cuenta se hace necesario el estudio de otra herramienta que permita establecer comparaciones.

1.4.2 Sistema de Gestión Administrativa Escolar SIGAEWEB

El sistema de Gestión Administrativa Escolar (SIGAEWEB), es un producto informático integrado, provisto por el Ministerio de Educación de la Provincia de Santa Fe, Argentina (SIGAEWEB, 2018). Es una aplicación web disponible en internet diseñada para empleados en gestión administrativa en colegios. Permite la integración de la escuela y el Ministerio de Educación en un único sistema facilita la reducción sustancialmente de los tiempos requeridos para la obtención y procesamiento de información (SIGAEWEB, 2013). Las posibilidades que brinda son la gestión de alumnos, carreras, secciones, plazas, inasistencias, calificaciones, servicios alimentarios e infraestructura.

SIGAEWEB cuenta con un apartado de configuraciones que permite parametrizar algunos aspectos del sistema. Según (SIGAEWEB, 2010) los parámetros ajustables son:

- **Perfiles:** Se pueden definir diferentes perfiles de usuarios (director, secretario, preceptor, docente, entre otros...) especificando los permisos de acceso a cada módulo.
- **Grados y divisiones:** Se registran los grados y divisiones que existen en la escuela.
- **Espacios curriculares:** Se registran las asignaturas o espacios curriculares que se dictan en cada grado y división.

- **Libros de temas:** Se pueden definir los libros de temas utilizados para el registro de calificaciones.
- **Otras:** Se pueden gestionar elementos como el escudo e identificación de la escuela, tipo de documentos aceptados, días hábiles e inhábiles, entre otros...

Las tecnologías con las que fue desarrollada esta herramienta son PHP 7 con el *framework* Laravel en el *backend*, Javascript, HTML Y CSS en el desarrollo del lado del cliente y para la conexión con la base de datos, el sistema utiliza PostgreSQL. Cuenta con una arquitectura web por lo que se puede acceder desde cualquier navegador. Por último cuenta con servicios web desarrollados en SOAP y REST, lo que permite la integración con otros sistemas.

Los beneficios de esta herramienta van desde la disminución de las tareas administrativas, reducción de los tiempos de respuesta en los trámites iniciados por los directores de escuelas y los docentes, así como la disponibilidad *online* de información de alumnos, secciones y docentes, indicadores pedagógicos y comensales (Santa Fe Provincia, 2022).

Algunas de las principales desventajas que presenta este sistema es la dependencia total de conexión a internet para su uso, en caso de que la conexión a Internet sea deficiente o inestable, el acceso a la herramienta podría verse afectado. Actualmente se ha visto afectado su funcionamiento por la interrupción de soporte a nivel central, ya que solo se realiza soporte a nivel de institución, lo que ha provocado que presente vulnerabilidades de seguridad. Por último no es capaz de adaptar su comportamiento y contenido a diferentes centros educativos con características particulares. A partir de los referentes internacionales que se han estudiado, se requiere el análisis de herramientas que a nivel nacional realicen estas funciones.

1.4.3 Distra

Los Sistemas de Planeación de Recursos Empresariales o por sus siglas en inglés ERP son sistemas diseñados para aprovechar las potencialidades que ofrecen las tecnologías de la información y las comunicaciones (TICs) en la integración de todos los subsistemas. Distra es un sistema de gestión empresarial, que permite integrar y optimizar los procesos empresariales, manejar con eficacia la información y disminuir los costos de las operaciones (XETID, 2023). Este producto lleva a digital todas las áreas de gestión: la logística, el inven-

tario, lo relacionado con la gestión de proyectos, recursos humanos, planificación financiera, y de actividades (MUÑOA, 2022). Es un sistema cubano desarrollado en la XETID empresa de referencia destinada al desarrollo de tecnologías de la información para la defensa.

Ofrece un conjunto completo de funciones básicas que responden a los dominios de solución de gestión organizacional, planificación estratégica y material, gestión del capital humano, contabilidad y finanzas, gestión de las relaciones con el cliente, ventas, logística, que atienden las necesidades de gestión empresarial (Oraldo Jacinto Simón, 2016). Según (Reyes, 2022) las prestaciones de este sistema permiten que desde los hogares los trabajadores puedan acceder al software RP-DISTRA, disponible en la red de redes. El auge de su utilización fue durante el año 2020, donde la COVID-19 afectó la asistencia a los centros de trabajo por lo que se explotó en el país el teletrabajo.

Distra cuenta con un módulo de “Datos Maestros” que tiene como objetivo posibilitar la configuración general de elementos que serán utilizados en el resto de los subsistemas del ERP. Permite gestionar los ejercicios y períodos que serán utilizados por las entidades para el registro de sus operaciones contables. Posibilita la gestión de las fechas por subsistemas que serán empleadas por la entidad para garantizar el control de las operaciones. Gestiona los formatos y niveles que definen la estructura de los componentes del sistema. Por último, brinda la posibilidad de manejar las monedas que serán utilizadas para las operaciones por entidad. A pesar de contar con este apartado, y con módulos independientes, no es posible modificar la estructura del sistema sin que los especialistas de la XETID realicen un estudio previo. A partir de un análisis de las necesidades de la institución se propone la distribución más adecuada y la adaptación del software para ella. Esto significa un atraso para la entrega del producto final al cliente.

El gobierno cubano aboga por el uso de herramientas de software libre, lo que constituye un pilar en los lineamientos para el desarrollo del software. Durante el desarrollo de este sistema se utilizó PHP para la lógica del negocio y Javascript para la vista del cliente, para la gestión de las bases de datos PostgreSQL (Oraldo Jacinto Simón, 2016). Tomando Distra como refe-

rente nacional, es posible llegar a ciertas conclusiones sobre la actualidad de estas plataformas en el mercado.

1.4.4 Análisis del estudio realizado a los módulos de configuración de soluciones existentes.

Con el fin de establecer comparaciones se presenta a continuación una breve descripción de criterios, estos fueron definidos por los autores de la presente investigación:

- **Adaptabilidad de sus componentes desde la interfaz de usuario (UI):** Es la característica que le permite la modificación de su estructura a la aplicación para adaptarla a diferentes entornos sin necesidad de modificación de la misma.
- **Disponibilidad Web:** Es la posibilidad de acceder a través de internet a la plataforma.
- **Control y gestión de elementos:** Es la característica del sistema que permite la gestión de elementos de cualquier tipo.
- **Tecnología utilizada:** Lenguajes de programación utilizados.
- **Seguimiento:** Es la realización de mantenimientos regulares por parte de los desarrolladores.

Teniendo esto en cuenta, se presenta una tabla resumen donde se recogen los datos más significativos de cada una de las soluciones homólogas estudiadas:

Tabla 1. Características generales de las soluciones estudiadas. Fuente: Elaboración propia.

SOLUCIONES CRITERIOS	SISTEMA INTEGRADO DE GESTIÓN ADMINISTRATIVA	SIGAEWEB	DISTRA
Adaptabilidad de sus componentes desde UI	No	No	No

Disponibilidad Web	Si	Si	Si
Control y gestión de elementos	Si	Si	Si
Tecnologías utilizadas	C# Javascript PL/SQL	PHP Javascript PostgreSQL	PHP JavaScript PostgreSQL
Seguimiento	No	No	Si

Teniendo en cuenta el estudio de las soluciones anteriormente descritas arroja las siguientes conclusiones:

- Estos sistemas no responden a la actualidad de la Dirección de Informatización de la Universidad de las Ciencias Informáticas.
- Es necesario que el módulo de configuración del Sistema XABAL SIGA Versión 2.0 permita la selección de sus componentes desde la interfaz de usuario.
- Es importante dar seguimiento y mantenimiento al sistema con el fin de evitar vulnerabilidades y problemas de seguridad.
- La referencia nacional Distra utiliza herramientas de software libre, comportamiento que debe seguir el Sistema de Gestión de Alimentación en su versión 2.0.
- Todos los sistemas utilizan JavaScript como lenguaje de programación para el lado del cliente.

A partir de estos elementos se precisa la realización de un estudio para la definición de las tecnologías y herramientas necesarias en el desarrollo del módulo de configuración XABAL SIGA Versión 2.0.

1.5 Estudio de herramientas y tecnologías.

Realizar un estudio de las tecnologías antes de comenzar un proyecto permite tomar decisiones informadas y estratégicas que maximizan las posibilidades de éxito del proyecto (Mirza, 2019). Ayuda a seleccionar las herramientas y tecnologías adecuadas que se ajusten a los requisitos, mejoren la eficiencia, sean escalables y sostenibles, se integren bien con otros sistemas y sean compatibles con los recursos disponibles.

Para el desarrollo del módulo de configuración de este sistema se tienen en cuenta dos partes fundamentales: *frontend* o lado del cliente y *backend* o lado del servidor. El *frontend* en programación se refiere a la parte de un sitio web o aplicación que los usuarios interactúan directamente, a través de la interfaz gráfica de usuario por sus siglas en inglés (GUI) (H. Al-Rababah, 2018) El *frontend* o desarrollo del lado del cliente se refiere también a la práctica de producir HTML, CSS y JavaScript. Estos tres elementos se encargan de dar forma a la parte frontal de un sitio web o aplicación. Esto incluye los fondos, colores, texto, animaciones o efectos (Coppola, 2023). Por otro lado, el *backend* es la parte de una aplicación web que se encarga de procesar los datos y la lógica de negocio en el lado del servidor (M. Alam, 2017).

En este epígrafe se realizará un estudio con el fin de seleccionar y describir cada una de las herramientas y tecnologías que serán utilizadas tanto en el desarrollo del lado del cliente como en el desarrollo del lado del servidor.

1.5.1 Lenguajes de programación para el desarrollo del lado del cliente.

Las tecnologías para el desarrollo del lado del cliente desempeñan un papel fundamental en la creación de aplicaciones web modernas (McNeil, 2020). Estas tecnologías se centran en la interfaz de usuario y la experiencia del mismo, permitiendo la creación de interfaces atractivas, interactivas y adaptables. Además, estas permiten desarrollar aplicaciones rápidas y eficientes, optimizando el rendimiento y la velocidad de la aplicación. También facilitan la adaptabilidad de la aplicación a diferentes dispositivos y tamaños de pantalla, brindando una experiencia consistente en múltiples plataformas. Asimismo, permiten la integración con servicios externos, enriqueciendo la funcionalidad de la aplicación y mejorando su utilidad.

Según los últimos datos de *GitHub Octoverse*, JavaScript se posiciona en el año 2022 como el lenguaje de programación más utilizado del mundo para desarrollo del lado del cliente por los profesionales de la plataforma (GitHub Octoverse, 2022). Este es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas (Amazon, 2023). Fue creado originalmente para agregar interactividad a las páginas web y se ha convertido en uno de los lenguajes más utilizados (Flanagan, 2020). Una de las características más destacadas de JavaScript es su capacidad de adaptarse y evolucionar rápidamente. Además según el estudio realizado en esta investigación de las soluciones similares existentes, en los tres casos se utilizó este lenguaje. Teniendo esto en cuenta se utiliza este lenguaje en el desarrollo del lado del cliente del módulo de configuración del sistema XABAL SIGA en su versión 2.0, aunque esto unido a un *framework* de desarrollo que se determinará en el siguiente apartado.

1.5.2 Frameworks de desarrollo del lado del cliente.

Los *frameworks* de desarrollo para el lado del cliente son herramientas de *software* que proporcionan una estructura y conjunto de funcionalidades para facilitar la creación de aplicaciones web interactivas y dinámicas (A.Ademola & T.Ademola, 2019). Partiendo de la selección de Javascript como lenguaje y según encuesta realizada por Stack Overflow en el año 2022 los tres más utilizados son React.js, Angular y Vue.js (Stackoverflow, 2022). La encuesta se realizó a desarrolladores profesionales de todo el mundo y se basó en la cantidad de ellos que afirmaron haber utilizado cada uno de estos *frameworks* en los últimos doce meses. A continuación se realiza un estudio y se detalla cada uno de ellos.

Para comenzar se presenta React JS, es una biblioteca Javascript para crear interfaces de usuario (REACT, 2023). La interfaz de usuario se construye a partir de pequeñas unidades como botones, texto e imágenes. React tiene una curva de aprendizaje media y es conocido por su flexibilidad en términos de elección de herramientas y arquitectura. Desde sitios web hasta aplicaciones de teléfonos, todo en la pantalla se puede descomponer en componentes. Esta tecnología utiliza un modelo de programación basado en componentes, lo que significa que los desarrolladores pueden construir aplicaciones web a partir de pequeñas piezas de código reutilizable (REACT DOCUMENTATION, 2023). Esto facilita la construcción y el man-

tenimiento de aplicaciones web complejas. En cuanto al rendimiento, React es altamente eficiente, y su arquitectura unidireccional contribuye a una experiencia de usuario fluida.

Por otro lado, Angular es un *framework* de desarrollo web de Google (Angular, 2023). Está basado en *TypeScript*, un *superset* de JavaScript que agrega tipos estáticos al lenguaje, lo que mejora la calidad y la robustez del código. Angular tiene una curva de aprendizaje medianamente pronunciada debido a su estructura y conceptos complejos, pero esta curva se compensa con su capacidad para gestionar aplicaciones web grandes y complejas de manera eficiente. Ofrece una gran flexibilidad en términos de arquitectura y estructura de la aplicación, lo que permite una mayor modularidad y reutilización de código (Angular, 2023). Angular es un *framework* completo que proporciona todo lo necesario para desarrollar aplicaciones web, incluyendo una CLI poderosa. En términos de rendimiento, Angular ha mejorado significativamente en versiones recientes y ofrece un rendimiento sólido.

Por último, Vue.js es un *framework* de JavaScript progresivo y fácil de aprender. Está basado en JavaScript y es conocido por su curva de aprendizaje baja, lo que lo hace ideal para desarrolladores principiantes. Vue.js es extremadamente flexible y se puede utilizar como una biblioteca para pequeñas partes de una aplicación o como un *framework* completo para proyectos más grandes (Vue.js, 2023). Vue.js ofrece un excelente rendimiento gracias a su sistema de reactividad y a su capacidad para optimizar automáticamente la actualización de la interfaz de usuario. Sin embargo, a pesar de sus ventajas, existen algunos inconvenientes que deben tener en cuenta debido a su relativa juventud cuenta con menos recursos y comunidad de soporte en comparación con los otros dos *frameworks* propuestos

Para resumir el estudio realizado sobre estos *frameworks* de desarrollo, se realiza la siguiente tabla con sus características principales teniendo en cuenta los criterios de Petukhova (2019) :

Tabla 2. Características generales de las tecnologías estudiadas. Fuente: Elaboración propia.

Criterios Tecnologías	Curva de aprendizaje	Flexibilidad	Tipo	Lenguaje	Rendimiento
React.js	Media	Alta	Biblioteca	Javascript	Alto
Angular	Media-alta	Alta	Framework	Javascript	Alto
Vue.js	Media-baja	Alta	Framework	Javascript	Alto

Para el desarrollo del Módulo de Configuración se utilizará React.js ya que es eficaz, solo se actualizan los componentes que han cambiado, permite la reutilización de componentes lo que significa que los desarrolladores pueden construir aplicaciones web a partir de fragmentos de código reutilizables. React puede ser beneficioso para el desarrollo de aplicaciones web debido a su eficiencia, reutilización de componentes, flexibilidad, comunidad activa y capacidad de desarrollar aplicaciones móviles multiplataforma (REACT, 2023). A continuación se realiza un estudio de otras herramientas utilizadas en el desarrollo del lado del cliente.

1.5.3 Otras tecnologías utilizadas en el desarrollo del lado del cliente.

Es importante mencionar que, si bien JavaScript desempeña un papel fundamental en el desarrollo del lado del cliente, no es el único lenguaje necesario para crear aplicaciones web. HTML y CSS también son componentes esenciales para construir sitios web modernos y funcionales.

HTML5 es un estándar que sirve como referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado HTML) para la definición de contenido de una página web, como texto, imágenes, vídeos, juegos, entre otros (Jiménez, 2019). Además propone estándares para cada aspecto de la web y un propósito claro para cada una de las tecnologías involucradas. Cada parte del documento está diferenciada, declarada y determinada por etiquetas específicas, aunque para

el lado del cliente se hace necesario contar con apariencias amigables que sean placenteras al usuario por lo que se utiliza también en este desarrollo CSS.

CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas (Miguel Angel Alvarez, 2017). El objetivo final es que los desarrolladores sean capaces de aplicar estilos a los documentos HTML de manera versátil y facilitar la separación entre contenido y presentación.

Una vez se han definido todas las tecnologías para el desarrollo del lado del cliente se procede al estudio de las tecnologías para la selección de las más adecuadas para el desarrollo del lado del servidor.

1.5.4 Lenguajes de programación para el desarrollo del lado del servidor.

Los lenguajes de programación para el desarrollo del lado del servidor son aquellos que se utilizan para crear y gestionar la lógica y la funcionalidad del servidor en una aplicación web (Briseño, 2021). Se ejecutan en el servidor y son responsables de procesar las solicitudes del cliente, acceder a bases de datos, realizar operaciones de procesamiento de datos y enviar las respuestas al cliente. Para la selección del lenguaje de programación a utilizar para el desarrollo del lado del servidor es preciso consultar información sobre los más utilizados, con mayor comunidad y perspectiva a futuro. Para ello estudiaremos el ranking de lenguajes de programación de Redmonk.

Redmonk es una empresa que se especializa en la evaluación y clasificación de tecnologías de software y lenguajes de programación (RedMonk, 2023). La compañía es conocida por su informe semestral de clasificación de lenguajes de programación, llamado "*Redmonk Programming Language Rankings*", que evalúa la popularidad y el uso de los lenguajes de programación en la comunidad de desarrollo. Sus clasificaciones y análisis ayudan a los desarrolladores y empresas a tomar decisiones sobre qué tecnologías utilizar en sus proyectos. En el artículo "Clasificación del lenguaje de programación RedMonk: enero de 2023"

(O`Grady, 2023) se define a JavaScript como el lenguaje más popular, seguido por Python, Java y PHP.

El índice de popularidad es importante a la hora de la selección de las tecnologías por varios factores, el primero es que generalmente las más populares cuentan con una mayor comunidad, lo cual es beneficioso en términos de soporte, documentación y tutoriales. Además mientras más popular son tienden a tener mayor estabilidad y madurez. Aunque este factor no debe ser único a la hora de realizar una selección, en este caso permite centrar la investigación en las tres tecnologías más populares. Teniendo en cuenta que se utiliza JavaScript en el desarrollo del lado del cliente, es preciso estudiar como posibles propuestas los lenguajes Python, Java y PHP para el desarrollo del lado del servidor.

Como primer lenguaje a estudiar, se propone Python el cual es un lenguaje de programación de alto nivel que permite realizar algunas tareas complejas en pocas líneas de código y de manera legible. (Ivet Challenger-Pérez, 2014). Una de las ventajas del uso de este, es su facilidad de uso y aprendizaje, lo que lo hace una buena opción para principiantes en programación. En efecto, cuenta con una amplia variedad de bibliotecas y módulos disponibles, lo que lo hace muy versátil y útil para diversas aplicaciones. La sintaxis de Python es tan sencilla y cercana al lenguaje natural que los programas elaborados en él parecen pseudocódigo. Python puede adaptarse a diversos proyectos y requerimientos y mantener alto rendimiento ante grandes cargas de información. Por último cuenta con una elevada reputación en términos de seguridad. Además, el hecho de contar con una gran comunidad activa permite que las fisuras de seguridad sean rápidamente solucionadas.

Por otro lado se presenta Java, el cual es un lenguaje de programación ampliamente utilizado para codificar aplicaciones web. Posee una curva de aprendizaje media debido a su enfoque en la programación orientada a objetos, su sintaxis rigurosa y la necesidad de más líneas de código para lograr una funcionalidad (Raul Gabriel Urma, 2020). Java cuenta con una sólida comunidad de desarrollo y amplio soporte tanto de Oracle como de la comunidad. Se sustenta de una documentación detallada y una gran cantidad de recursos en línea para ayudar a los desarrolladores a resolver problemas y aprender nuevas técnicas. Presenta un

buen rendimiento y escalabilidad a la hora de trabajar con aplicaciones de alta carga. Además, cuenta con características incorporadas para garantizar la seguridad de las aplicaciones, como el manejo de excepciones y la gestión de memoria. La comunidad de desarrollo de Java trabaja activamente en la identificación y resolución de vulnerabilidades de seguridad, lo que contribuye a la confiabilidad y la protección de las aplicaciones Java.

Por último, se presenta *Hypertext Preprocessor* (PHP) que es un lenguaje de código especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML (PHP, 2023). PHP es conocido por ser relativamente fácil de aprender aunque su sintaxis está diseñada para aquellos que tienen experiencia previa en programación. A pesar de presentar una gran flexibilidad, puede llevar a una mayor variabilidad en el estilo de codificación y a la falta de coherencia en el código. Por último, a medida que las aplicaciones web crecen en tamaño y complejidad, PHP puede encontrar limitaciones en términos de escalabilidad (Zandstra, 2021). PHP tiene un rendimiento inferior que otras tecnologías, especialmente en aplicaciones de alta carga. En cuanto a seguridad ha mejorado su enfoque con el tiempo, pero históricamente ha sido más propenso a vulnerabilidades debido a la flexibilidad y las malas prácticas de programación (Zandstra, 2021).

Una vez planteada una visión global sobre estos lenguajes es preciso realizar comparaciones directas entre ellos con el fin de seleccionar el que más se ajusta a las características de este proyecto. Para ello se definen a continuación los parámetros seleccionados:

- **Sintaxis y legibilidad:** Se refiere a cómo se estructura y se presenta el código de un lenguaje de programación, y cómo se puede entender y leer fácilmente.
- **Soporte y comunidad:** Se refiere a la disponibilidad de recursos y ayuda técnica para los desarrolladores y a la red de desarrolladores, entusiastas y usuarios que utilizan un lenguaje de programación específico.
- **Rendimiento:** El rendimiento se refiere a la eficiencia y velocidad con la que una aplicación ejecuta sus tareas y maneja altas cargas de trabajo.

- **Seguridad:** Hace referencia a las medidas y prácticas implementadas para proteger la integridad, confidencialidad y disponibilidad de los datos y recursos involucrados.

Teniendo conocimiento acerca de los parámetros necesarios para realizar las comparaciones se presenta la siguiente tabla:

Tabla 3. Comparación entre lenguajes de programación para desarrollo del lado del servidor. Fuente: Elaboración propia.

Lenguaje / Criterio	Python	Java	PHP
Sintaxis y legibilidad	Fácil sintaxis y legibilidad.	Sintaxis más estricta y rigurosa.	Fácil sintaxis y legibilidad
Soporte y comunidad	Gran comunidad y continuo soporte.	Gran comunidad y continuo soporte.	Gran comunidad y continuo soporte.
Rendimiento	Gran rendimiento con grandes volúmenes de información.	Gran rendimiento con grandes volúmenes de información.	Rendimiento medio para grandes volúmenes de información.
Seguridad	Elevada reputación en términos de seguridad.	Sólida reputación en términos de seguridad	Actualmente cuenta con una buena seguridad

Para el desarrollo del lado del servidor del módulo de configuración para el sistema XABAL SIGA en su versión 2.0 se escoge Python. El hecho de contar con un gran rendimiento para aplicaciones de alto volumen de información es un aspecto fundamental en este caso, ya que el sistema a desarrollar se plantea llegar a la mayor cantidad de clientes posibles, para lo cual es esencial contar con un adecuado rendimiento. Por otro lado su fácil sintaxis y legibilidad potencia el hecho de que el equipo de desarrollo cuenta con conocimientos previos de este lenguaje. Además de la seguridad que brinda con vistas al futuro por la existencia de gran comunidad y documentación. Definido el lenguaje es preciso seleccionar que *framework* de desarrollo sería más adecuado para el módulo de configuración.

1.5.5 Frameworks de desarrollo web para el lado del servidor.

Los *frameworks* de desarrollo para *backend* son una colección de bibliotecas y herramientas que proporcionan un conjunto de funcionalidades para el desarrollo de aplicaciones web, como la gestión de bases de datos, la autenticación de usuarios, la seguridad, la gestión de sesiones, el manejo de errores y la generación de API (N. K. Sharma, 2015). Teniendo en cuenta el estudio realizado previo a este epígrafe, para el desarrollo del módulo de configuración de XABAL SIGA en su versión 2.0 se utilizará Python como lenguaje en la programación. A partir de esto, los tres *frameworks* de desarrollo web en Python más populares son Flask, Django y FastAPI, según estudio realizado por A. Singh y M. Bansal (A. Singh, 2018).

Django Rest Framework es un paquete potente y flexible para Django que permite crear APIs REST de forma rápida y sencilla (Chillarón, 2016). Dentro de las características principales que posee se encuentra la serialización que permite la conversión de datos en formato JSON o XML para ser utilizados en la interfaz de programación de aplicaciones (API). Además proporciona una variedad de vistas predefinidas para manejar operaciones CRUD (Crear, Leer, Actualizar y Eliminar) en los recursos de la API, permite la autenticación de usuarios y la autorización de acceso a los recursos de la API. Es un framework que cuenta con una amplia documentación y apoyo de la comunidad.

Para establecer comparaciones y determinar la tecnología que más se ajusta a las características del proyecto, se analiza Flask. Este es un *framework* de desarrollo web en Python que se utiliza para construir aplicaciones web escalables y seguras. Es ligero y se centra en la simplicidad y la flexibilidad (N. K. Sharma, 2015). Una de sus características distintivas es su arquitectura "micro". Se centra en proporcionar un conjunto básico de herramientas y funcionalidades para la creación de aplicaciones web, y permite a los desarrolladores agregar funcionalidades adicionales según sea necesario a través de extensiones y módulos de terceros. (FLASK, 2022).

Como tercera opción se ofrece FastAPI un *framework* de desarrollo web que se ha vuelto muy popular en la comunidad de Python diseñado para crear APIs de manera eficiente (FastAPI, 2023). FastAPI se basa en la programación asíncrona para lograr una alta veloci-

dad de procesamiento (Pydantic, 2023). Esto significa que puede manejar fácilmente cargas de trabajo pesadas y escalables, lo que lo convierte en una excelente opción para aplicaciones de alto rendimiento. Un aspecto negativo es que si no estás familiarizado con la sintaxis de la programación asíncrona en Python, puede hacer que requiera una curva de aprendizaje inicial mayor. Además debido a que es relativamente nuevo y está en constante evolución, puede haber menos compatibilidad con versiones anteriores en comparación con *frameworks* más establecidos. Esto puede requerir actualizaciones y ajustes en el código existente al migrar a nuevas versiones.

Tanto FastAPI, Flask como Django son *frameworks* de desarrollo web en Python muy capaces que pueden ser utilizados para crear sistemas de gestión web con múltiples módulos y un módulo de configuración. Un estudio realizado por Adeyemo (A. A. Adeyemo, 2019) determina que Django es una mejor opción para aplicaciones empresariales a gran escala debido a su capacidad integrada para manejar la autenticación de usuarios, la administración de datos y la gestión de formularios .

Teniendo en cuenta el estudio realizado previo, para el desarrollo del módulo de configuración para el Sistema de Gestión de Alimentación se hace necesario *frameworks* que faciliten la creación de *APIs Restful*. Por lo cual las diferencias entre FastAPI y Django son las determinantes a la hora de seleccionar una tecnología. El poco tiempo de FastAPI en el mercado hace que sea muy inseguro con vistas al futuro, ya que puede cambiar sus políticas y licencias y esto es fundamental para el desarrollo. Es imprescindible para la universidad y el país que todo desarrollo se realice con herramientas de código abierto con el objetivo de preservar la soberanía tecnológica del país. Por otro lado FastAPI cuenta con una menor comunidad activa, lo cuál puede ser perjudicial para la solución de errores y vulnerabilidades de la tecnología.

Django, además es altamente personalizable y se integra fácilmente con otras bibliotecas y *frameworks* de Python, lo que lo hace muy útil para el desarrollo de aplicaciones web complejas. Por último este *framework* de desarrollo está diseñado para manejar grandes cantidades de datos y solicitudes, lo que lo hace muy útil para aplicaciones web con alta demanda.

(Django Rest Framework, 2022). A partir de lo planteado, se decide utilizar Django Rest Framework para el desarrollo. Es preciso entonces realizar un estudio para la selección del sistema gestor de bases de datos que utilizará el sistema.

1.5.6 Sistema gestor de bases de datos

Una base de datos es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos (Marqués, 2011). Partiendo de esto, en la actualidad existen sistemas de bases de datos y gestores. Un sistema de bases de datos es un conjunto de herramientas y software que permite a los usuarios almacenar, acceder y modificar grandes cantidades de datos de manera eficiente (R. Elmasri, 1994). Por otro lado, un gestor de bases de datos es un software que se utiliza para crear y administrar bases de datos, y para permitir a los usuarios acceder a los datos almacenados en ellas (J. C. Cubero, 2012). Estos gestores permiten optimizar el tiempo y facilitar al equipo de desarrollo el trabajo con las bases de datos de la aplicación.

Teniendo en cuenta la encuesta realizada por Stackoverflow (2022), el sistema de bases de datos más utilizado en el mundo es PostgreSQL y el gestor de bases de datos PgAdmin. Tomando esto como un estudio de referencia en el mercado actual, la gran comunidad que presentan estas tecnologías y el estudio de las soluciones similares existentes, se utilizarán para esta parte del desarrollo. A continuación se detalla cada una de ellas.

PostgreSQL es un potente sistema de base de datos relacionales de objetos de código abierto que utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas. Se ha ganado una sólida reputación por su arquitectura probada, integridad de datos, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para ofrecer constantemente soluciones innovadoras y de alto rendimiento. No es sorprendente que PostgreSQL se haya convertido en la base de datos relacional de código abierto más utilizada por desarrolladores y organizaciones (The PostgreSQL Global Development Group, 2023). En esta investigación se utilizó con el objetivo de almacenar la información con la que trabaja el módulo de configuración.

Por otro lado, PgAdmin es una herramienta gráfica de código abierto para administrar y desarrollar bases de datos en PostgreSQL. Esta herramienta está diseñada para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en varios lenguajes y para varios sistemas operativos (Ricardo Ruiz, 2015). Para el desarrollo del módulo de configuración se utilizó con el objetivo de trabajar con los objetos que están almacenados en la base de datos, examinar sus propiedades y realizar tareas administrativas.

Ahora bien, una vez se ha definido sistema de base de datos y gestor, es preciso determinar cuáles son las herramientas que se utilizarán con el fin de facilitar el trabajo en el desarrollo del módulo de configuración para el sistema XABAL SIGA versión 2.0.

1.5.7 Otras herramientas

Para el desarrollo y modelación de la propuesta de solución a la problemática planteada se utilizan algunas herramientas con el fin de facilitar el desarrollo. Este tipo de tecnologías van desde un editor de código fuente potente, como un software para modelación de UML hasta un sistema para controlar las versiones. A continuación, se describen cada una de ellas.

En primer lugar se utiliza Visual Studio Code (VSC) que es un editor de código fuente ligero, pero potente que se ejecuta en escritorio. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes como C++, C#, Java, Python, PHP, Go, Django (Visual Studio, 2023). VSC incluye un depurador interactivo, por lo que puede recorrer el código fuente, inspeccionar variables, ver pilas de llamadas y ejecutar comandos en la consola. También se integra con herramientas de compilación y secuencias de comandos para realizar tareas comunes y acelerar los flujos de trabajo diarios. VS Code es compatible con Git, por lo que puede trabajar con control de código fuente sin salir del editor, incluida la visualización de diferencias de cambios pendientes (Visual Studio, 2023).

Otra herramienta utilizada es el Visual Paradigm la cual es una herramienta UML que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos,

construcción, pruebas y despliegue. Ayuda a una rápida construcción de aplicaciones de gran calidad, mejoras y a un menor costo de producción. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Está diseñado para desarrollar software con Programación Orientada a Objetos, permite reducir la duración del ciclo de desarrollo brindando ayuda a los arquitectos, analistas, diseñadores y desarrolladores. Busca también automatizar tareas tediosas que pueden distraer a los desarrolladores. Para el desarrollo del módulo de configuración se utilizó con el objetivo de modelar la base de datos y esquemas del negocio.

Por último se utiliza también Git, sistema de control de versiones distribuido utilizado principalmente en el desarrollo de software. Un sistema de control de versiones es una herramienta que permite gestionar los cambios realizados en los archivos de un proyecto a lo largo del tiempo (Westby, 2015). Git registra y almacena todos los cambios realizados en los archivos, lo que permite rastrear y revertir cambios, colaborar con otros desarrolladores de manera eficiente y mantener un historial completo de todas las modificaciones. Utilizar esto para este proyecto facilita en caso de surgir problemas o errores, se pueda retroceder a versiones anteriores para identificar y solucionar los problemas. Además, en proyectos como el presente donde se trabaja en equipo, varias personas pueden hacerlo en los mismos archivos simultáneamente. Por estos elementos es que se utiliza esta herramienta en el desarrollo. Para continuar con la investigación es prudente realizar un estudio para definir que metodología de desarrollo es la adecuada en este proyecto.

1.6 Metodología

Para desarrollar el módulo de configuración destinado a XABAL SIGA 2.0 se hace necesario seleccionar las metodologías que mejor se adapten a los requerimientos que debe tener la plataforma. Estas guías son herramientas que permiten a desarrolladores realizar un proyecto efectivo mediante el seguimiento de preceptos, que varían según las características de cada equipo de trabajo.

Las metodologías de desarrollo de software son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático (Universidad de

Santander, 2020). El objetivo de las distintas metodologías es el de organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible. Las metodologías han evolucionado de manera significativa en las últimas décadas, tanto así, que pueden permitir el éxito o el fracaso de muchos de los sistemas desarrollados para distintas áreas (Arencibia, 2018).

Según (Nekaka, 2020) en la actualidad se pueden mencionar dos grandes grupos de metodologías de desarrollo de software las tradicionales y las ágiles. Las metodologías tradicionales que son aquellas que ponen mayor énfasis en la planificación y control del proyecto y en la especificación precisa de requisitos y modelado. Por otro lado las ágiles están más orientadas a la generación de código con ciclo muy corto de desarrollo, se dirigen a equipos pequeños, priorizan los recursos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso.

Según el criterio de (Mory, 2021) la metodología tradicional es predictiva y rígida, considera el proyecto como un todo y suele tener poca comunicación con el cliente, entregando el *software* al final con documentación extensa. En contraste, la metodología ágil es adaptativa y flexible, descompone el proyecto en partes más pequeñas, mantiene una comunicación constante con el cliente y realiza entregas periódicas de software, minimizando la documentación. La ingeniería de software ágil representa una alternativa razonable a la ingeniería de software convencional para ciertas clases de software y en algunos tipos de proyectos. A partir de esto, se selecciona la metodología ágil para el desarrollo del módulo de configuración para XABAL SIGA Versión 2.0 ya que permite al equipo adaptarse a los cambios y necesidades del proyecto en tiempo real. Los ciclos de entrega para el cliente son cortos, y debido a que es un desarrollo para un plazo de tiempo medio-corto, se ajusta a la perfección. Al incluir al cliente en el proceso permite detectar tempranamente errores y fallos en la aplicación. Las tres metodologías ágiles para desarrollo de software más utilizadas en el mundo son Scrum, Kanban y Extreme Programming (XP) (Sotomayor, 2021).

Programación Extrema (XP) pone el énfasis en: la satisfacción del cliente y en la entrega rápida de software incremental, los equipos pequeños y muy motivados para efectuar el proyecto, los métodos informales, los productos del trabajo con mínima ingeniería de software y la sencillez general en el desarrollo (Pressman, 2010). La planeación en el desarrollo ágil implica escuchar al cliente, crear historias de usuario y asignarles valores y costos. El diseño sigue el principio de "mantenerlo sencillo" y se guía por la implementación de las historias de usuario (Pressman, 2010).

La programación en parejas es una técnica clave para la actividad de codificación, lo que permite la solución de problemas en tiempo real y el aseguramiento de la calidad. El código se integra continuamente para evitar problemas de compatibilidad y descubrir errores a tiempo. La estrategia de "integración continua" proporciona un ambiente de "prueba de humo" para validar el código (Pressman, 2010). En XP, es importante crear pruebas unitarias automatizadas antes de la codificación para permitir pruebas de regresión y asegurar la calidad del software. Las pruebas se organizan en un "grupo de prueba universal" para permitir pruebas diarias de integración y validación del sistema. Las pruebas de aceptación XP, especificadas por el cliente, se enfocan en la funcionalidad visible y son derivadas de las historias de usuario implementadas en la liberación del software. Gran parte de la eficacia de este modelo de planificación deriva de una división clara de responsabilidades, que tiene en cuenta las necesidades del negocio en todo momento (Trujillo, 2007).

Por otro lado, la metodología de desarrollo Scrum la cual se basa en ciclos de trabajo llamados "*sprints*" y enfatiza la colaboración y la adaptabilidad (Sutherland, 2020). Scrum es especialmente efectivo para proyectos donde los requisitos pueden cambiar o no están completamente definidos desde el principio. Sus elementos principales incluyen roles definidos, como el *scrum master*, el propietario del producto y el equipo de desarrollo, así como artefactos como la lista priorizada de características y el Incremento. Además, los llamados sprints, tienen una duración fija y conducen a entregas incrementales de producto (Schwaber, 2023). Los valores de Scrum, como el compromiso y la transparencia, impulsan la colaboración y la

adaptabilidad (Scrum Alliance, 2023). En este sentido se permite entregas frecuentes de software parcialmente entregable y una comunicación constante con el cliente.

En tercer lugar, la metodología Kanban la cual es una metodología ágil que se destaca por su enfoque en la visualización y gestión eficiente del flujo de trabajo (Kanban University, 2023). Utiliza un tablero Kanban para representar visualmente tareas y limita la cantidad de trabajo en curso para mantener un flujo constante. No define roles específicos y se adapta fácilmente a diferentes flujos de trabajo. Kanban se basa en la mejora continua, permitiendo a los equipos analizar métricas y ajustar su proceso en tiempo real, lo que lo hace adecuado para la gestión de proyectos y la optimización de procesos (Anderson, 2010).

En este caso se selecciona XP para el proyecto ya que se enfoca en la entrega temprana y frecuente de software funcional y valorado por el cliente. Además, favorece la colaboración estrecha entre el equipo de desarrollo y el cliente, lo que facilita la comprensión de las necesidades del mismo y la adaptación rápida a los cambios en los requisitos. XP también promueve la calidad del software a través de la automatización de pruebas, la programación en parejas y el rediseño disciplinado.

1.7 Conclusiones del capítulo.

Durante el desarrollo del capítulo se arriba a las siguientes conclusiones parciales:

- Se definieron las bases para el desarrollo, identificando debilidades y fortalezas del diagnóstico del módulo de configuración actual.
- Se seleccionó como entorno de desarrollo: Python unido a Django Rest Framework, React.JS y PostreSQL utilizando la metodología XP.
- Se determinaron estándares que debe cumplir el módulo de configuración mediante el estudio de soluciones homólogas.

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se detalla la propuesta de solución, estableciendo los principales componentes a tener en cuenta y los usuarios para los que está destinado partiendo del estudio del arte realizado. Además se describe el procedimiento seguido en la etapa de planificación definiendo historias de usuario y los elementos necesarios para definir un orden de prioridades y tiempos de trabajo y entrega. Por último se expone la fase de diseño propuesta por la metodología XP donde se definen todos los elementos relacionados con el diseño de la solución propuesta: diseño arquitectónico, patrones de diseño y tarjetas CRC.

2.1 Propuesta de solución.

Para dar respuesta al objetivo de esta investigación, luego de analizar las necesidades del sistema y realizar la selección de las herramientas para la implementación, se definen los elementos a desarrollar. El módulo de configuración del sistema XABAL SIGA versión 2.0 debe contar con las configuraciones de abastecimiento, cajero, facturación, distribución, reservación, configuración y seguridad. En la siguiente imagen se describen las configuraciones que debe contener el módulo:



Figura 2. Propuesta de solución. Fuente: Elaboración propia.

Como se muestra en la Figura 2. Propuesta de solución. Fuente: Elaboración propia. y en comparación con lo representado en la Figura 1. Componentes del módulo de configuración actual del sistema XABAL SIGA. Fuente: Elaboración propia., el módulo de configuración propuesto mantiene la misma estructura básica aunque con algunos cambios significativos los cuales se describen a continuación:

- **Arquitectura:** Se propone un cambio de arquitectura que permita una gran escalabilidad de datos sin tener que crear instancias de la aplicación. Además poder realizar la distribución del software como un servicio.
- **Tecnología:** Se propone realizar un cambio de tecnología basado en un estudio detallado del mercado, teniendo en cuenta que estas deben satisfacer las necesidades de la Dirección de Informatización de la Universidad de las Ciencias Informáticas. Esta propuesta viene encaminada por las vulnerabilidades actuales que presenta el sistema. Un cambio de tecnología también optimiza el rendimiento de la aplicación, además ofrece mejoras de escalabilidad y mantenibilidad.

- **Cambio de Moneda:** Se propone que todos los precios de la aplicación sean expresados en Moneda Nacional y se elimine el CUC que ya no circula en el país.
- **Abastecimiento:** En este apartado se propone unificar la gestión de clasificaciones de unidades de medida con la gestión de unidades de medida y la unificación de los tipos de producto con las categorías de tipo de producto, evitando configuraciones redundantes.
- **Facturación:** Se propone realizar la configuración de cobro, que antes se hacía en Configuración con el fin de ordenar mejor los ajustes de la aplicación y la eliminación de las reglas a excluir.
- **Distribución:** Se propone la unificación de Rangos de evento con Evento, con el objetivo de agrupar funcionalidades redundantes y la eliminación de la configuración de reglas.
- **Reservación:** Se propone añadir la configuración del proceso de reservación, donde se brinda la posibilidad de que el sistema funcione con o sin reservación. Esto debido a que el sistema debe satisfacer diferentes necesidades de clientes, los cuales pueden tener este requerimiento o no.
- **Configuración:** Se propone adicionar las configuraciones de gestión de área y gestión de tipos de áreas, con el fin de que cada institución pueda administrar sus áreas y tipos de áreas.
- **Seguridad:** Se propone la adición de un gestor de instituciones para satisfacer la característica principal del sistema y convertirlo en un sistema multiinstitucional y versátil con componentes que puedan ser seleccionables en dependencia de las necesidades del cliente.

Todas estas propuestas buscan solucionar los problemas actuales que presenta el módulo de Configuración del Sistema XABAL SIGA y darle cumplimiento al objetivo general de esta

investigación. Una vez ha sido definida una propuesta de solución, se procede a realizar un análisis sobre los usuarios que intervienen y operan el módulo de configuración.

2.2 Usuarios

Según el libro “Desarrollo de Software: Requisitos, Análisis y Diseño” publicado por la editorial de la Universidad de La Plata en Argentina, los usuarios de un sistema son aquellas personas o entidades que interactúan con él para cumplir con objetivos específicos (Olsina L., 2016). Son personas que tienen diferentes roles o perfiles, tales como usuarios finales, administradores, desarrolladores y otros. Cada uno de estos roles tiene diferentes necesidades y requerimientos en cuanto a la funcionalidad del sistema. En la siguiente tabla se definen los usuarios del módulo de configuración de XABAL SIGA Versión 2.0:

Tabla 4. Descripción por usuario del sistema. Fuente: Elaboración propia.

Usuario	Descripción
Administrador general	Tiene permisos para acceder a todos los apartados del módulo de configuración.
Administrador institucional	Tiene acceso a todos los apartados del módulo de configuración que involucran su institución.

Una vez han sido definidos los usuarios que tendrán acceso al módulo de configuración se procede a la fase de planificación según indica la metodología XP.

2.3 Fase 1: Planificación

En la fase de planificación del proyecto se define el alcance general del mismo. Durante la planificación de la iteración, el equipo de desarrollo y el cliente trabajan juntos para establecer los objetivos de la iteración y los plazos para su cumplimiento (Salazar, 2019). El equipo de desarrollo utiliza la información obtenida durante la fundamentación teórica para determinar qué historias de usuario se incluirán en la próxima iteración y qué tareas son necesarias para completarlas. El cliente y el equipo de desarrollo también acuerdan qué funcionalidades son prioritarias y cuáles pueden esperar para iteraciones futuras.

2.3.1 Historias de Usuario (HU).

Las historias de usuario son una breve descripción de una necesidad del usuario que se escribe en el lenguaje del cliente final (Beck, 2019). Las HU son la técnica que utiliza XP para especificar los requisitos de software (Díaz, 2016). En esta fase, el equipo de desarrollo y el cliente trabajan juntos para seleccionar las historias de usuario que se implementarán durante la próxima iteración y para establecer los objetivos y plazos de la iteración.

Es importante destacar que en el caso de este proyecto, con el objetivo de optimizar la eficiencia en el desarrollo y mantener la coherencia del código, se tomó la decisión de agrupar las historias de usuario (HU) relacionadas con las operaciones CRUD. CRUD es un acrónimo que se refiere a las operaciones básicas que se pueden realizar en una base de datos o en cualquier sistema informático que permita la gestión de datos. Las letras de CRUD representan las palabras en inglés *Create, Read, Update y Delete* (Smith, 2018). A esta agrupación de operaciones, en esta investigación se les denomina como gestionar y en este caso se añade el Buscar. Esta decisión se fundamenta en varios aspectos clave, en primer lugar, al abordar las operaciones CRUD como un todo coherente, es posible implementar y entregar rápidamente la funcionalidad completa. En lugar de trabajar en pequeñas partes individuales, además permite aprovechar las características y herramientas proporcionadas por Django Rest Framework para simplificar el desarrollo de las operaciones CRUD. Como resultado, se ahorra tiempo y esfuerzo en comparación con la creación de historias de usuario separadas para cada operación. En esta investigación se les llama a los CRUD gestionar.

A continuación se definen los aspectos fundamentales que deben incluir las HU para el proyecto de desarrollo según (Beck, 1999):

- **Número:** Es el número asignado a cada HU.
- **Nombre:** Atributo que posee el nombre la de HU.
- **Usuario:** Es el usuario del sistema que utiliza o protagoniza el requisito que se describe.
- **Prioridad en negocio:** Evidencia el nivel de prioridad de la HU en el negocio.

- **Riesgo en desarrollo:** Evidencia el nivel de riesgo en caso de no realizarse.
- **Puntos estimados:** Atributo que contiene la estimación hecha por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es uno equivale a una semana ideal de trabajo de lunes a viernes. En la metodología XP está definida una semana ideal como cinco días hábiles, trabajando cuarenta horas semanales.
- **Iteración asignada:** Precisa la iteración en la que será desarrollada la HU.
- **Descripción:** Posee una breve descripción de lo que realizará la HU.
- **Observaciones:** Brinda información extra que se estime agregar para hacer más comprensible la HU.

Una vez se han especificado los elementos necesarios, y siguiendo con los parámetros anteriormente descritos, se plantea a continuación dos ejemplos de las HU definidas para llevar a cabo el desarrollo del módulo de configuración, para ver el resto visitar el Anexo 1.

Historias de usuario del módulo de configuración del sistema XABAL SIGA versión 2.0.:

Tabla 5. Historia de usuario 1 Gestionar clasificación de platos. Fuente: Elaboración propia.

Historia de usuario	
Número: 1	Nombre: Gestionar clasificación de platos
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.4	Iteración asignada: 1
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
Descripción: <ul style="list-style-type: none"> • Muestra una lista de clasificaciones de platos, con el nombre, si está activo y las opciones de Registrar, Editar y Eliminar clasificación de plato, además de la posibilidad de buscar y filtrar. • Permite cambiar el paginado y la cantidad de registros por página. 	

- Registrar clasificación de plato: se muestra una interfaz con los campos: nombre, activo y descripción, acompañado de los botones aceptar y cancelar.
- Editar: se muestra interfaz con los campos: nombre, activo y descripción, acompañado de los botones aceptar y cancelar.
- Eliminar: se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:

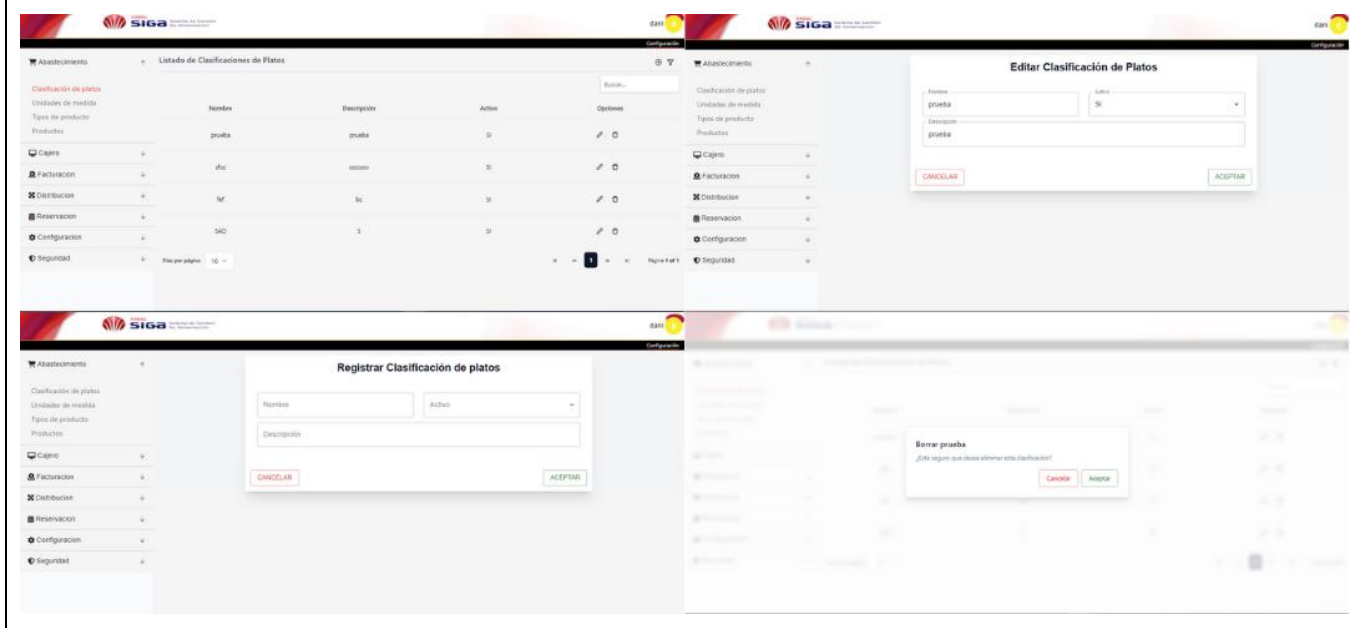


Tabla 6. Historia de usuario Gestionar unidades de medida. Fuente: Elaboración propia.

Historia de usuario	
Número: 2	Nombre: Gestionar unidades de medida
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 0.4	Iteración asignada: 1

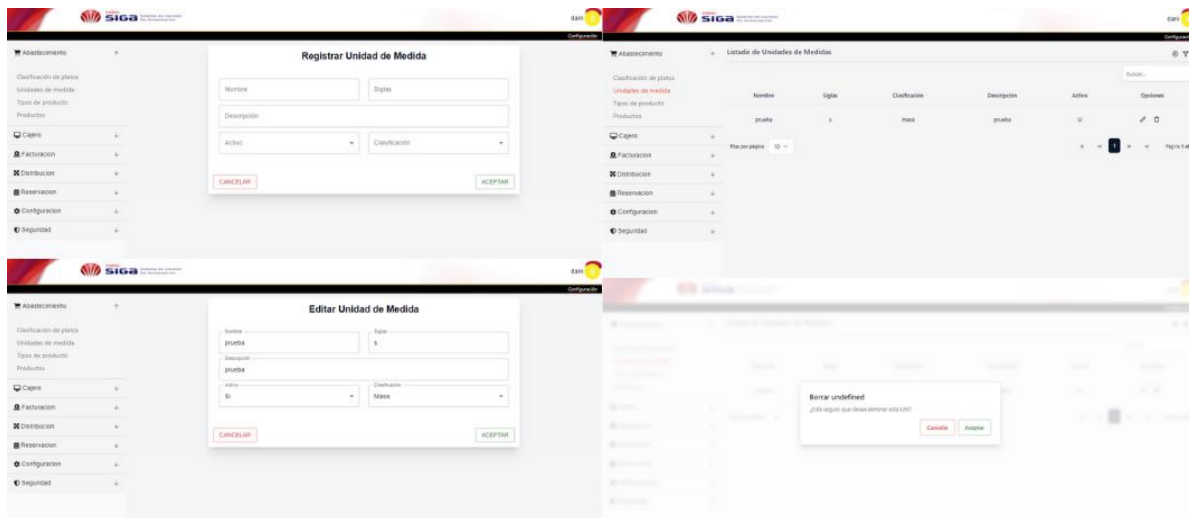
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

Descripción:

- Muestra una lista de unidades de medida, con el nombre, siglas, clasificación, descripción, si está activo y las opciones de Registrar, Editar y Eliminar. Además de la posibilidad de buscar y filtrar por activo, clasificación y nombre.
- También permite cambiar el paginado y la cantidad de registros por página.
- Registrar unidad de medida: se muestra una interfaz con los campos: nombre, siglas, clasificación, activo y descripción, acompañado de los botones aceptar y cancelar.
- Editar unidad de medida: se muestra interfaz con los nombre, siglas, clasificación, activo y descripción, acompañado de los botones aceptar y cancelar.
- Eliminar unidad de medida: se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



Teniendo en cuenta las historias de usuario, es preciso definir los requisitos no funcionales del sistema.

2.3.2 Requisitos no funcionales.

Los requisitos no funcionales de un sistema son características o atributos que no están directamente relacionados con las funciones específicas que realiza, sino que se centran en aspectos más generales (Pons, 2017). A continuación se especifican los requisitos no funcionales del módulo de configuración:

RnF de usabilidad:

- El módulo no requiere conocimientos técnicos para poder utilizarlo, ya que todo está fácil de entender y con iconos sugerentes.
- El módulo debe permitir acceder a diferentes interfaces con una profundidad máxima de 2 clics.
- El módulo debe presentar una interfaz simple para facilitar el manejo a los usuarios.

RnF de interfaz o apariencia externa:

- El módulo debe contar con interfaces donde predominen los colores rojo, gris, blanco, negro y tonalidades de naranja y amarillo.
- El módulo debe contar con una interfaz de usuario adaptable a diferentes dispositivos y tamaños.
- El módulo debe contar con interfaces con una letra legible y tamaño preferiblemente 11.

RnF de *software*:

- El módulo debe funcionar con navegador web Mozilla Firefox superior a la versión 50.0 y Google Chrome superior a la versión 50.0.
- El módulo debe funcionar en cualquier distribución de SO Linux y Windows Seven o superior.
- El módulo requiere una conexión a internet estable.

RnF de *hardware*:

- El módulo requerirá al ordenador al menos 1Gb de RAM y un procesador de dos núcleos de al menos 1.6 GHz.

RnF de rendimiento:

- El módulo debe responder a las transiciones en no más de 800 milisegundos.

Una vez se han definido todas las historias de usuario y los requisitos no funcionales del sistema es preciso establecer una estimación de esfuerzos por cada HU.

2.3.3 Estimación de esfuerzos por HU

La estimación de esfuerzos por Historias de Usuario (HU) es un proceso importante para planificar y controlar el desarrollo de software. La estimación de esfuerzos por HU se utiliza para determinar la cantidad de tiempo y recursos necesarios para implementar cada HU (Beck, Extreme Programming Explained: Embrace Change, 2019). Esta estimación se basa en la complejidad relativa de cada HU en comparación con otras HU. Para el desarrollo de esta propuesta de solución se planifican las historias de usuario teniendo en cuenta que una semana equivale a cinco días laborables, tomando solamente de cada día 4 horas para desarrollo. En la siguiente tabla se muestra la estimación de esfuerzos por HU correspondientes a esta investigación:

Tabla 7. Estimación de esfuerzos por HU. Fuente: Elaboración propia.

Número	Historia de Usuario	Estimación (Semanas)
1	Gestionar clasificación de platos	0,4
2	Gestionar unidades de medida	0,4

3	Gestionar tipos de productos	0,5
4	Gestionar productos	0,5
5	Gestionar tipos de tarjetas	0,4
6	Gestionar tarjetas	0,4
7	Asociar tarjeta a usuario	0,4
8	Gestionar asignación de IP a puerta	0,5
9	Configurar cantidad de accesos	0,4
10	Gestionar torpedos	0,5
11	Administrar solapín perdido	0,4
12	Gestionar áreas	0,5
13	Gestionar categorías	0,5
14	Gestionar estructuras	0,5
15	Configurar estructura	0,4
16	Gestionar eventos	0,4
17	Gestionar horarios	0,4
18	Gestionar tipos de áreas	0,5
19	Administrar períodos de reservación	0,3
20	Administrar elementos a mostrar	0,3
21	Administrar datos de contacto	0,3
22	Administrar responsables de áreas	0,5

23	Gestionar módulos por institución	0,5
24	Gestionar instituciones	0,4
25	Administrar configuraciones del proceso	0,6
26	Gestionar configuración de cobro	0,4
27	Administrar configuración de comensales	0,6
28	Administrar comensales	0,5
29	Gestionar permisos	0,5
30	Gestionar roles	0,5
31	Gestionar usuarios	0,6
32	Autenticación de usuarios	1

La estimación total del esfuerzo fue de 15 semanas. Tomando esto en consideración se procede a realizar un análisis de plan de iteraciones.

2.3.4 Plan de iteraciones.

El plan de iteraciones es un documento que describe las historias de usuario seleccionadas, los objetivos y las fechas límite para la próxima iteración (Beck, 2019). El plan de iteraciones se crea durante la fase de planificación de la iteración, en la que el equipo de desarrollo y el cliente trabajan juntos para seleccionar las historias de usuario que se implementarán durante la próxima iteración. Una vez que se han seleccionado las historias de usuario, se establecen los objetivos de la iteración y se determinan las fechas límite para la entrega del software. Para el desarrollo de esta propuesta de solución se planifican las historias de usuario teniendo en cuenta que una semana equivale a cinco días laborables, tomando solamente de cada día 4 horas para desarrollo. A continuación se presenta una tabla con la estimación por semanas para la entrega de cada iteración.

Tabla 8. Plan de iteraciones. Fuente: Elaboración propia.

Iteración	Historia de Usuario	Estimación (Semanas)
1	Gestionar clasificación de platos	10
	Gestionar unidades de medida	
	Gestionar tipos de productos	
	Gestionar productos	
	Gestionar tipos de tarjetas	
	Gestionar tarjetas	
	Gestionar asignación de IP a puerta	
	Gestionar torpedos	
	Gestionar áreas	
	Gestionar categorías	
	Gestionar estructuras	
	Gestionar eventos	
	Gestionar horarios	
	Gestionar módulos por institución	
	Gestionar instituciones	
	Gestionar configuración de cobro	
	Gestionar permisos	
	Gestionar roles	
	Gestionar usuarios	
Autenticación de usuarios		
Gestionar tipos de áreas		
2	Asociar tarjeta a usuario	5
	Configurar cantidad de accesos	
	Administrar solapín perdido	
	Configurar estructura	
	Administrar períodos de reservación	
	Administrar elementos a mostrar	
	Administrar datos de contacto	
	Administrar responsables de áreas	
	Administrar configuraciones del proceso	
	Administrar configuración de comensales	

	Administrar comensales	
--	------------------------	--

La separación de historias de usuario por iteraciones se realiza teniendo en cuenta los principios de entrega rápida y continua de funcionalidades valiosas al cliente (Beck, Extreme Programming Explained: Embrace Change, 2019). Teniendo en cuenta esto, para el desarrollo del módulo de configuración del Sistema de Gestión de Alimentación Versión 2.0 se realizarán 2 iteraciones. Una primera iteración donde se entregará la parte más importante del sistema, con una duración de 10 semanas de desarrollo y una segunda iteración con el resto de los requisitos del cliente con una duración de 5 semanas. Esto establece una duración total de desarrollo de 15 semanas.

La duración de cada iteración en la metodología Extreme Programming (XP) puede estar condicionada por varios factores, y uno de ellos es el tamaño del equipo de desarrollo y su nivel de conocimiento de las tecnologías utilizadas (Beck, Extreme Programming Explained: Embrace Change, 2019). En el caso del desarrollo del módulo de configuración para el Sistema XABAL SIGA Versión 2.0 se cuenta con un equipo pequeño de desarrollo con conocimientos limitados de las tecnologías y herramientas por lo que en el tiempo establecido de cada iteración se tiene en cuenta estos elementos.

2.3.5 Plan de entregas

El plan de entregas establece las HU que serán agrupadas para conformar una entrega del sistema y el orden en que se realizarán. A diferencia de los enfoques tradicionales de desarrollo de software, XP se centra en la entrega incremental y frecuente de software funcionando. El cronograma de entregas se analiza sobre la base de las estimaciones de tiempos de desarrollo. A continuación se presenta una tabla con las iteraciones con su fecha de entrega:

Tabla 9. Fechas de entregables. Fuente: Elaboración propia.

Entregable	Iteración1	Iteración 2
Fecha inicio	26-06-2023	4-09-2023

Fecha de entrega	4-09-2023	9-10-2023
------------------	-----------	-----------

Una vez realizado el plan de entregas concluye la primera fase de planificación y comienza segunda fase, el diseño del sistema.

2.4 Fase II: Diseño del sistema.

La fase de diseño en la metodología XP (Extreme Programming) es una etapa crucial del proceso de desarrollo de software que se centra en la creación de un diseño sólido y coherente que se adapte a las necesidades del usuario y del negocio. Para comenzar con el diseño del sistema es preciso definir una arquitectura de *software*.

2.4.1 Diseño arquitectónico.

Según (Sommerville, 2011) en su libro “Ingeniería de Software” la arquitectura de *software* es la estructura o estructuras del sistema, que comprenden elementos de *software*, las propiedades visibles externamente de esos elementos y las relaciones entre ellos. Teniendo presente los problemas antes descritos que posee actualmente el Sistema de Gestión de Alimentación XABAL SIGA 2.0 se propone un cambio de arquitectura. Los principales problemas que presenta son en cuanto a la escalabilidad, bajo rendimiento y tecnologías obsoletas.

Para el desarrollo del Sistema de Gestión de Alimentación XABAL SIGA 2.0 se propone una arquitectura de software como servicio, a partir de que se concibe como un sistema en la nube que podrá ser consumido por clientes con características propias de personalización. *Software as Service* (SaaS) es un modelo de servicios que ofrece la capacidad al consumidor de usar aplicaciones proporcionadas por el proveedor, las cuales se ejecutan sobre una infraestructura en la nube (Beatriz Elena Cano Maya, 2014).

Las aplicaciones son accedidas desde diferentes dispositivos cliente, a través de una interfaz ligera, como un navegador web o una interfaz de programa. El instituto nacional de estándares y tecnologías (NIST *National Institute of Standards and Technology*) define el desarrollo en la nube como un modelo que permite el acceso, a través de la red y bajo demanda, a un conjunto de recursos informáticos configurables; por ejemplo redes, servidores, almacenamiento,

aplicaciones y servicios, que pueden ser rápidamente aprovisionados y liberados con un esfuerzo mínimo de gestión e interacción con el proveedor de servicios (NIST, 2011). Teniendo todos estos elementos, a continuación se muestra el diseño arquitectónico:

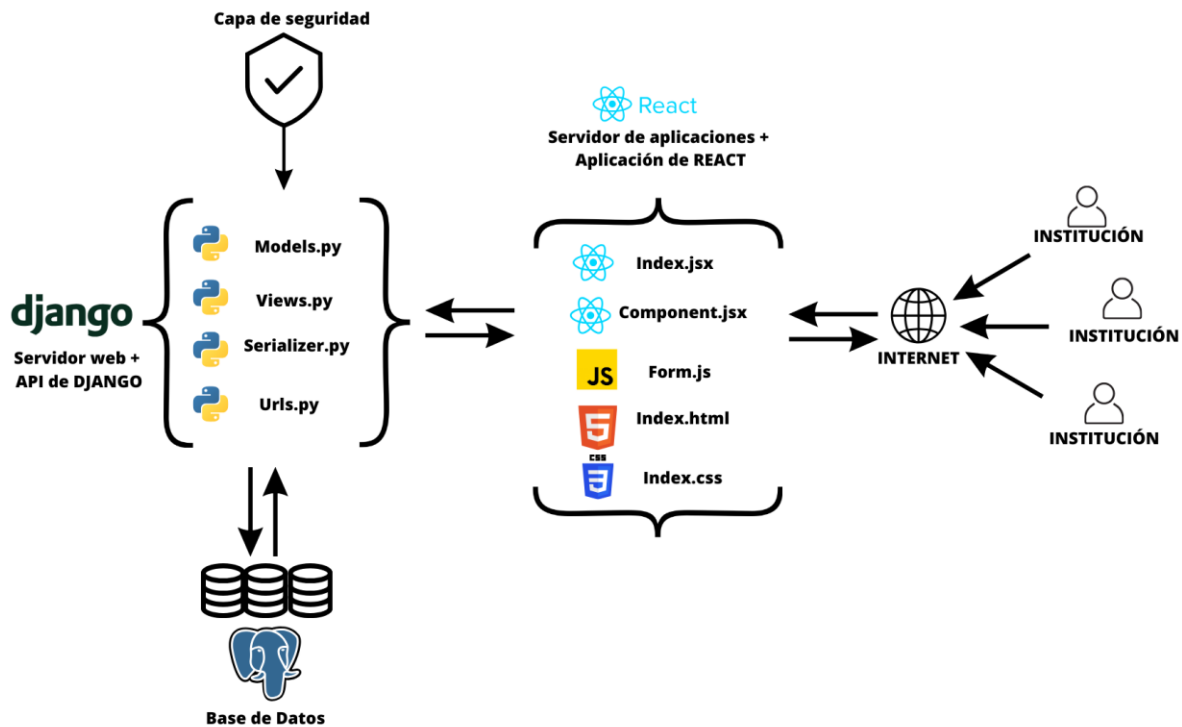


Figura 3. Diseño arquitectónico para XABAL SIGA Versión 2.0. Fuente: Elaboración propia.

Como indica la representación de la Figura 3. Diseño arquitectónico para XABAL SIGA Versión 2.0. Fuente: Elaboración propia., se plantea un diseño arquitectónico del sistema, basado en la arquitectura de *Software as Service* anteriormente descrita. Las flechas en el diagrama muestran el flujo de tráfico de datos y comunicación entre los diferentes componentes de la arquitectura. La aplicación se despliega en un servidor central que aloja la aplicación y la base de datos. Cada institución tiene acceso a sus propios datos a través de una interfaz de usuario personalizada. Los principales componentes del diagrama son:

Los componentes principales que componen la Figura 3. Diseño arquitectónico para XABAL SIGA Versión 2.0. Fuente: Elaboración propia. SON:

- **Servidor central:** Este servidor aloja tanto la aplicación como la base de datos y se comunica a través de la API de Django.
- **Clientes:** Cada cliente tiene acceso a sus propios datos a través de una interfaz de usuario personalizada que les permite acceder y visualizar sus propios elementos de manera segura.
- **Servidor web + API:** Este servidor aloja la API de Django y se comunica con la base de datos para realizar operaciones de lectura y escritura de datos. La API de Django es la interfaz entre React y la base de datos. La comunicación entre el servidor web y el servidor de aplicaciones se realiza a través de HTTP o HTTPS.
- **Servidor de aplicaciones + Aplicación de REACT:** Este servidor aloja la aplicación React, que se comunica con la API de Django a través de HTTP para obtener los datos necesarios para mostrarlos en la interfaz de usuario personalizada de cada institución. La comunicación entre el cliente y el servidor de aplicaciones se realiza a través de HTTP o HTTPS.
- **Capa de seguridad:** Esta capa es responsable de proteger la aplicación y los datos de los usuarios. Incluye autenticación y autorización de usuarios, encriptación de datos y protección contra ataques de seguridad como XSS, CSRF, *SQL Injection*, entre otros.
- **Base de datos:** La base de datos almacena todos los datos y se comunica con la aplicación a través de la API de Django.

Una vez se ha planteado el diseño de la arquitectura, es posible determinar los patrones de diseño que seguirá el equipo de desarrollo de software.

2.4.2 Patrones de diseño

Una de las prácticas clave de XP es el uso de patrones de diseño para mejorar la calidad del código y la eficiencia del equipo de desarrollo. Según Beck y Andrés (2004), los patrones de diseño son soluciones probadas a problemas comunes de diseño de software que se han identificado y documentado en forma de patrones. El uso de patrones de diseño permite a los

desarrolladores "aprender de la experiencia de otros y aplicarla a sus propios problemas de diseño". Los patrones ayudan a los desarrolladores a crear un diseño de software más modular, flexible y fácil de entender. Además, el uso de ellos también mejora la comunicación y la colaboración entre los miembros del equipo de desarrollo.

Los Patrones Generales de Asignación de Responsabilidades en Software (GRASP) son un conjunto de patrones de diseño de *software* que se utilizan para asignar responsabilidades en un diseño orientado a objetos. (Oviedo-García M. A., 2013). A continuación se detallan los patrones GRASP que se utilizaron en el desarrollo del módulo de configuración:

- **Creador:** se utiliza para asignar la responsabilidad de crear objetos a una clase específica. En el desarrollo de este sistema se utiliza en los serializadores.
- **Controlador:** es empleado para asignar la responsabilidad de controlar la lógica del sistema a una clase específica. Se utiliza en el desarrollo en las vistas.
- **Experto en Información:** se utiliza para asignar la responsabilidad de manejar la información a la clase que tiene la información necesaria para realizar la tarea. Para el desarrollo del módulo de configuración se utiliza en los modelos.

Los patrones GOF, también conocidos como *Gang of Four* (GoF), son un conjunto de patrones de diseño de software que son ampliamente reconocidos y utilizados en la comunidad de desarrollo de software para abordar problemas comunes de diseño y arquitectura de software (Erich Gamma, 1994). A continuación se detallan los patrones GoF que se utilizaron en el desarrollo del módulo de configuración:

- **Decorador:** se utiliza para añadir funcionalidades a objetos colocando estos objetos dentro de objetos encapsuladores especiales que contienen estas funcionalidades, en el desarrollo se utilizó en la documentación de una aplicación con Swagger, especialmente para describir y definir elementos de la API de una manera más organizada y reutilizable.

Siguiendo por esta línea de ideas una vez se han establecido los patrones de diseño es correcto definir las tarjetas Clase-Responsabilidad-Colaborador a lo que se dedica el siguiente epígrafe.

2.4.3 Tarjetas Clase - Responsabilidad – Colaborador (CRC).

La metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). En el siguiente epígrafe las historias de usuario se dividen en tareas donde cada una representa una característica distinta del sistema. Las tarjetas CRC están compuestas por los siguientes elementos:

- **Una clase:** representa una colección de objetos similares.
- **Una responsabilidad:** es aquello que la clase conoce o realiza. Una clase puede cambiar el valor de lo que sabe pero no puede cambiar el valor de lo que saben otras clases.
- **Las colaboraciones:** adoptan una de dos formas, pueden ser un pedido de información o una petición de que se realice una operación.

A continuación se presentan dos tarjetas CRC correspondientes a esta investigación, el resto se encuentran en el Anexo 2. Tarjetas CRC del módulo de configuración del sistema XABAL SIGA versión 2.0.:

Tabla 10. Tarjeta CRC Clasificación de platos .Fuente: Elaboración propia.

Nombre: TbNclasificaciondeplato	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear una nueva clasificación de platos.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de clasificación de platos.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar una clasificación de plato.</p> <p>update():Este método maneja la solicitud PUT para actualizar una clasificación de platos.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar parcialmente una clasificación de platos.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar una clasificación de platos.</p>	Institucion

Tabla 11. Tarjeta CRC Unidades de medida. Fuente: Elaboración propia.

Nombre: TbUnidadMedida	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear una nueva unidad de medida.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de unidades de medida.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar una unidad de medida.</p> <p>update():Este método maneja la solicitud PUT para actualizar una unidad de medida.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar parcialmente una unidad de medida.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar una unidad de medida.</p>	TbInstitucion models.Model

2.8 Conclusiones del capítulo

En este capítulo de la investigación se han trazado las bases para el desarrollo del sistema llegando a las siguientes conclusiones:

- Se establecieron las características y funcionalidades del sistema mediante la definición de historias de usuario.
- Se determinó un tiempo de duración de proyecto de 15 semanas.
- Se determinó realizar un cambio de arquitectura de software que propicie una buena organización a la hora de establecer una lógica de negocio.
- Se establecieron los patrones de diseño a utilizar durante el desarrollo.

Capítulo III: Implementación y pruebas del sistema

En este capítulo se define como objetivo describir la implementación y las pruebas realizadas en el sistema de gestión de alimentación, utilizando la metodología de Programación Extrema (XP). Además, se abordarán dos fases cruciales para el desarrollo exitoso del proyecto, la implementación, en la cual se llevarán a cabo las tareas de ingeniería para construir y configurar el módulo de configuración del sistema. Y por último, se abordará la fase de pruebas al sistema, donde se evaluarán exhaustivamente todas las funcionalidades implementadas, garantizando su calidad y correcto funcionamiento.

3.1 Fase III: Desarrollo

Esta fase de desarrollo en la Metodología de Programación Extrema (XP) representa un punto crucial en el ciclo de vida del proyecto de desarrollo de *software* (Beck, Extreme Programming Explained: Embrace Change, 2019). Durante esta etapa, se transforman las especificaciones y diseños iniciales en un producto funcional, utilizando una serie de prácticas y principios. Como primer elemento se definen las tareas de ingeniería que son necesarias para dar cumplimiento a las HU.

3.1.1 Tareas de ingeniería

En la metodología XP, las Historias de Usuario son una forma de describir los requisitos del software desde la perspectiva del usuario (Kent Beck, 2004). Estas pueden no ser entendibles para el cliente pero son de gran importancia ya que permiten definir cada una de las actividades que se llevan a cabo como parte del desarrollo de las HU. A continuación se exponen las tareas de ingeniería correspondientes a la Historia de Usuario Gestionar Clasificación de Platos, el resto de tareas de ingeniería pueden ser consultadas en el Anexo 3. Tareas de ingeniería por HU del módulo de configuración del sistema XABAL SIGA versión 2.0.:

Tabla 12. Tarea de ingeniería No.1 de la HU Gestionar clasificación de platos. Fuente: Elaboración propia.

Tarea	
Número de la Tarea: 1	Número de la Historia de Usuario: 1

Nombre de la tarea: Desarrollar vista para el gestionar clasificación de platos	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 26-06-2023	Fecha de fin: 27-06-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferrero Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de clasificación de platos. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) las clasificaciones de platos del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de clasificaciones de platos.	

Tabla 13. Tarea de ingeniería No.2 de la HU Gestionar clasificación de platos. Fuente: Elaboración propia.

Tarea	
Número de la Tarea: 2	Número de la Historia de Usuario: 1
Nombre de la tarea: Desarrollar funcionalidades para el gestionar clasificación de platos	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 27-06-2023	Fecha de fin: 28-06-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferrero Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) las clasifica-	

ciones de platos. El objetivo principal es desarrollar el *backend* y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes sobre las categorías de platos. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar clasificaciones de platos en la base de datos.

Una vez han sido definidas las tareas de ingeniería se precisa la realización de pruebas al sistema para comprobar la calidad de la solución propuesta.

3.2 Fase IV: Pruebas.

Las pruebas de software son un proceso sistemático y controlado que tiene como objetivo evaluar y verificar un sistema o programa de software para identificar defectos, errores o problemas en su funcionamiento (Pressman, 2010). Estas pruebas se realizan con el fin de asegurarse de que el software cumple con los requisitos especificados y funciona de manera confiable en su entorno previsto. Las pruebas de software son una parte integral del ciclo de vida del desarrollo de software y pueden incluir diversos tipos de pruebas, como pruebas unitarias, pruebas de integración, pruebas de sistema, pruebas de aceptación y más. Estas pueden ser manuales o automatizadas, y se llevan a cabo para asegurar la calidad y la confiabilidad del software, detectar y corregir defectos y garantizar que el software cumpla con las necesidades y expectativas del cliente.

La importancia de estas actividades radica en garantizar la calidad del software, detectar problemas temprano y adaptarse rápidamente a los cambios (Bustamante, 2014). En XP, se enfatiza la automatización de pruebas, la colaboración con el cliente y la retroalimentación continua para entregar software de alta calidad de manera incremental. Lo que no debe faltar es una comunicación efectiva entre el equipo de desarrollo y el cliente, así como la ejecución constante de pruebas automatizadas para mantener la confiabilidad del software.

Las pruebas funcionales son una forma de determinar si una aplicación funciona como se espera (ZAPTEST, 2023). Estas pruebas se enfocan en evaluar si el software cumple con los requisitos y especificaciones establecidos, y no en cómo se produce el procesamiento.

Durante el desarrollo de este módulo se realizaron pruebas funcionales, unitarias y de aceptación, incluyendo las pruebas de seguridad. Las pruebas funcionales realizadas fueron enfocadas a verificar todas las funcionalidades del módulo. Por otro lado las pruebas de seguridad estuvieron enfocadas en verificar que las instituciones solo puedan acceder y modificar la configuración de los componentes que se le han habilitado.

Estas pruebas y medidas de seguridad permitieron garantizar que las instituciones puedan utilizar este módulo de manera segura y eficiente. En el siguiente apartado se documentan las pruebas recomendadas por la metodología XP: unitarias y de aceptación, estando estas basadas en las pruebas realizadas durante el desarrollo.

3.2.1 Pruebas unitarias.

Las pruebas unitarias son un tipo esencial de prueba de *software* que se enfoca en evaluar componentes individuales de código, como funciones o métodos, de manera aislada y automatizada (Martin, 2008). Su objetivo primordial es garantizar que cada unidad de código funcione sin errores y cumpla con sus especificaciones. Durante las pruebas unitarias, se aísla la unidad de código bajo prueba, se automatizan los casos de prueba, y se verifica su funcionamiento técnico y funcional, incluyendo pruebas de borde y condiciones límite. Estas pruebas son cruciales para mantener la calidad del software, se integran con prácticas de desarrollo continuo y se basan en herramientas y marcos de prueba específicos.

A continuación se muestra la realización de pruebas unitarias al modelo relacionado con la Historia de Usuario Gestionar clasificación de platos.

```

1  from django.test import TestCase
2  from .models import TbnClasificacionPlato
3  from autenticacion.models.entities.institucion import Institucion
4
5  class TbnClasificacionPlatoTestCase(TestCase):
6      def setUp(self):
7          # Crea una instancia de Institucion para enlazarla
8          self.institucion = Institucion.objects.create(
9              name="Institución de Prueba",
10             description="Descripción de la Institución de Prueba",
11             active=True,
12             active_modules=["Módulo1", "Módulo2"],
13         )
14
15         # Crea una instancia de TbnClasificacionPlato para pruebas
16         self.clasificacion_plato = TbnClasificacionPlato.objects.create(
17             id_institucion=self.institucion,
18             nombre_clasificacion_plato="Clasificación de Plato de Prueba",
19             activo=True,
20             descripcion_clasificacion_plato="Descripción de Clasificación de Plato de Prueba",
21             fecha_registro="2023-09-28", # Ajusta la fecha según tus necesidades
22         )
23
24         def test_creacion_de_clasificacion_plato(self):
25             # Verifica si la instancia se ha creado correctamente
26             self.assertEqual(self.clasificacion_plato.nombre_clasificacion_plato,
27                             "Clasificación de Plato de Prueba")
28             self.assertTrue(self.clasificacion_plato.activo)
29             self.assertEqual(self.clasificacion_plato.descripcion_clasificacion_plato,
30                             "Descripción de Clasificación de Plato de Prueba")
31
32         def test_actualizacion_de_clasificacion_plato(self):
33             # Realiza una actualización en la instancia
34             self.clasificacion_plato.nombre_clasificacion_plato = "Nueva Clasificación de Plato"
35             self.clasificacion_plato.save()
36
37             # Verifica si la actualización se realizó correctamente
38             updated_clasificacion_plato = TbnClasificacionPlato.objects.get(pk=self.clasificacion_plato.pk)
39             self.assertEqual(updated_clasificacion_plato.nombre_clasificacion_plato,
40                             "Nueva Clasificación de Plato")
41
42         def test Eliminacion_de_clasificacion_plato(self):
43             # Obtiene el ID de la clasificación de plato antes de eliminarla
44             clasificacion_plato_id = self.clasificacion_plato.id_clasificacion_plato
45             self.clasificacion_plato.delete()
46
47             # Verifica si la clasificación de plato se ha eliminado correctamente
48             with self.assertRaises(TbnClasificacionPlato.DoesNotExist):
49                 TbnClasificacionPlato.objects.get(id_clasificacion_plato=clasificacion_plato_id)

```

Figura 4. Pruebas unitarias al Gestionar clasificación de platos. Fuente: Elaboración propia.

Las pruebas unitarias ejecutadas al “TbNclasificacionPlato” arrojaron los siguientes resultados:

```
● Found 3 test(s).  
Creating test database for alias 'default'...  
System check identified no issues (0 silenced).  
...  
-----  
Ran 3 tests in 0.017s  
  
OK  
Destroying test database for alias 'default'...
```

Figura 5. Resultados de pruebas unitarias realizadas al Gestionar clasificación de platos. Fuente: Elaboración propia

Se realizaron dos iteraciones de prueba, en la primera presentaron problemas los siguientes modelos: TbNevento, TbDpersonalIPPuerta, TbDsolapinPerdido, TbNtipoTarjeta, TbDproducto y TbNhorario. En la segunda se obtuvo resultados satisfactorios para todos los modelos. A continuación se presenta un gráfico resumen con los resultados obtenidos:

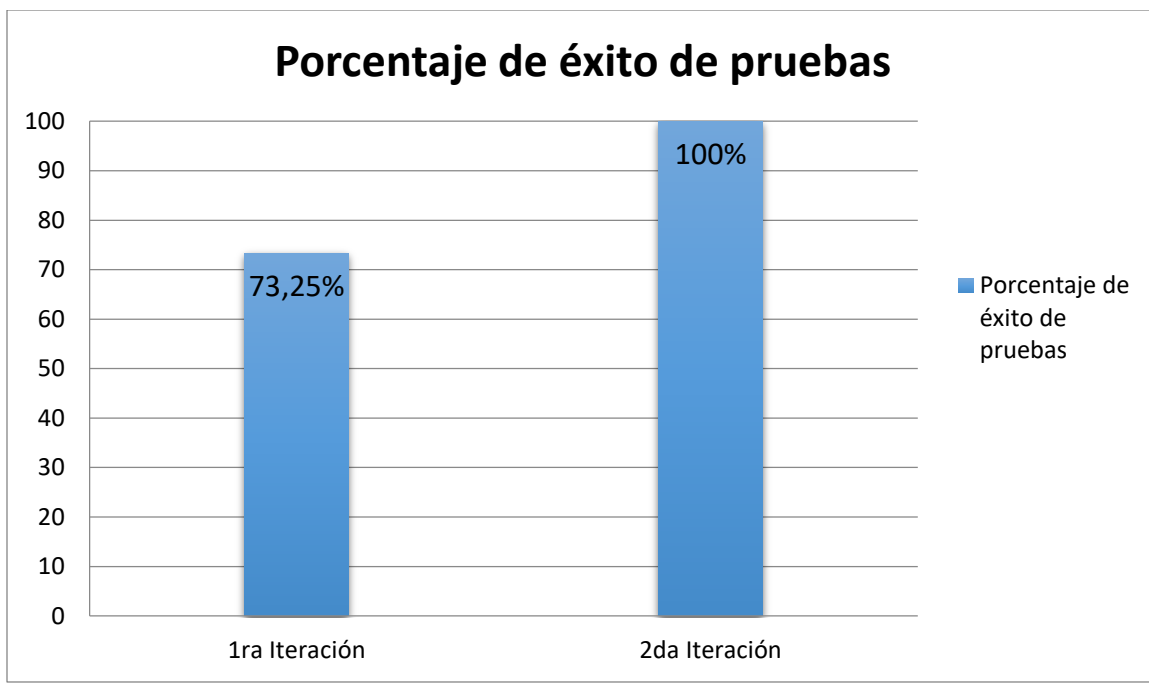


Figura 6. Comparativa de resultados de pruebas unitarias. Fuente: Elaboración propia.

Como resultado final se obtuvieron un total de 80 pruebas unitarias satisfactorias como se muestra en la siguiente figura:

```

TERMINAL  PORTS  GITLENS  COMMENTS  PROBLEMS  OUTPUT  DEBUG CONSOLE
-----
Ran 80 tests in 2.303s

OK
Destroying test database for alias 'default'...
    
```

Figura 7. Resultados finales de pruebas unitarias realizadas. Fuente: Elaboración propia.

Las pruebas realizadas arrojaron resultados satisfactorios y permitieron comprobar que las funciones de creación, edición y eliminación operan según lo planeado. Además, estas permitieron prevenir posibles fallos. El archivo completo de las pruebas unitarias puede ser consultado en el código de la aplicación.

3.2.2 Pruebas de aceptación.

Según la metodología XP las pruebas de aceptación son pruebas fundamentales para garantizar que la aplicación cumpla con los criterios de aceptación definidos por el cliente (Kent Beck, 2004). Estas pruebas se ejecutan regularmente, con la participación activa del cliente en la validación. Si las pruebas de aceptación revelan que la funcionalidad no cumple con los criterios, se realizan ajustes y se vuelven a ejecutar las pruebas hasta que se cumplan. Esto garantiza que el software entregado cumpla con las expectativas y necesidades del cliente, promoviendo la entrega de software de alta calidad de manera iterativa.

A continuación se describen las pruebas realizadas a la HU Gestionar clasificación de platos y Gestionar unidades de medida. El resto de pruebas pueden ser consultadas en el Anexo 4 Pruebas de aceptación al módulo de configuración del sistema XABAL SIGA Versión 2.0.

Tabla 14. Prueba de aceptación del Gestionar clasificación de platos. Fuente: Elaboración propia.

Caso de prueba de aceptación	
Código:H1_PA1	Historia de Usuario:1
Nombre: Gestionar clasificación de platos	
Descripción: Prueba de las funcionalidades adicionar, eliminar, modificar y buscar clasifi-	

cación de platos.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona abastecimiento.
2. Presiona clasificación de platos
3. Se carga la vista con el listado de clasificaciones de platos

Adicionar:

1. Presiona en Adicionar clasificación de platos.
2. Se muestra una interfaz con los campos: nombre, activo y descripción.
3. Se llenan los campos.
4. Presiona botón Aceptar.
5. Muestra una notificación con el mensaje "Clasificación de platos creada".

Eliminar

1. Presiona el botón Eliminar
2. Muestra que desea una interfaz con el mensaje "¿Está seguro eliminar esta clasificación?"
3. Presiona el botón Aceptar
4. Muestra una notificación con el mensaje "Clasificación de platos eliminada"

Modificar

1. Presiona en Editar clasificación de platos.
2. Se muestra una interfaz con los campos: nombre, activo y descripción.
3. Se llenan los campos
4. Presiona botón Aceptar
5. Muestra una notificación con el mensaje "Clasificación de platos editada"

Buscar

1. Llena el campo de búsqueda

2. Se filtran los elementos de la lista.
Resultado: Satisfactorio.

Tabla 15. Prueba de aceptación del Gestionar unidades de medida. Fuente: Elaboración propia.

Caso de prueba de aceptación	
Código:H2_PA2	Historia de Usuario:2
Nombre: Gestionar unidades de medida.	
Descripción: Prueba de las funcionalidades adicionar, eliminar, modificar y buscar unidades de medida.	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe estar autenticado • El usuario debe encontrarse en el módulo de configuración 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Presiona abastecimiento. 2. Presiona unidades de medida 3. Se carga la vista con el listado de unidades de medida <p>Adicionar:</p> <ol style="list-style-type: none"> 1. Presiona en Adicionar unidad de medida 2. Se muestra una interfaz con los campos: nombre, siglas, descripción, activo y clasificación. Seguido por los botones aceptar y cancelar. 3. Se llenan los campos 4. Presiona botón Aceptar 5. Se muestra una notificación con el mensaje “Unidad de medida creada”. <p>Eliminar</p> <ol style="list-style-type: none"> 1. Presiona el botón Eliminar 2. Se muestra una interfaz con el mensaje: “¿Está seguro que desea eliminar esta UM?” 3. Presiona el botón Aceptar 	

4. Muestra una notificación con el mensaje: “Unidad de medida eliminada”

Modificar

1. Presiona en Editar unidades de medida
2. Se muestra una interfaz con los campos: nombre, siglas, descripción, activo y clasificación. Seguido por los botones aceptar y cancelar.
3. Se llenan los campos
4. Presiona botón Aceptar
5. Muestra una notificación con el mensaje: “Unidad de medida editada”

Buscar

1. Llena el campo de búsqueda
2. Se filtran los elementos de la lista

Resultado: Satisfactorio.

En total se realizaron un total de 32 pruebas de aceptación en dos iteraciones, las cuales fueron llevadas a cabo con la participación activa de la Dirección de Informatización de la Universidad de las Ciencias Informáticas. En la primera iteración se encontraron problemas con algunos requisitos, pero en la segunda se obtuvo un ciento por ciento de aceptación. El correcto resultado de las mismas facilitó la entrega por parte del cliente de un aval de satisfacción, el cual puede ser consultado en el Anexo 5: Aval de satisfacción de cliente. Fuente: Dirección de Informatización. Este documento tiene como propósito expresar su conformidad con el software entregado.

3.3 Conclusiones del capítulo.

En este capítulo se especificó el proceso de implementación del módulo a partir del desglose de las HU en tareas de ingeniería. Esto permitió:

- Se especificaron los procedimientos necesarios para el desarrollo de las historias de usuario definidas.
- Se comprobó la calidad de la solución propuesta mediante la realización de pruebas unitarias y de aceptación a la misma.

CONCLUSIONES FINALES

Con el desarrollo de la presente investigación se arriba a las siguientes conclusiones:

- Se desarrolló un módulo de configuración para el sistema XABAL SIGA en su versión 2.0 adaptable a las necesidades de cada cliente.
- Las funcionalidades que fueron añadidas en la solución propuesta demostraron su utilidad y calidad.
- La arquitectura y tecnologías utilizadas fueron elementos clave en el éxito de la solución desarrollada.
- Se obtuvo un aval de satisfacción de cliente que demuestra el éxito de la solución propuesta.

RECOMENDACIONES

A partir de los resultados obtenidos se recomienda:

- Continuar con el desarrollo de los módulos de Abastecimiento, Reservación, Distribución, Cajero, Facturación y Reportes del sistema para la versión 2.0, a fin de lograr un sistema como un todo que pueda ser comercializable.
- Desarrollar una versión móvil para brindar una mejor experiencia al usuario.

REFERENCIAS BIBLIOGRÁFICAS

- A. A. Adeyemo, D. O. (2019). Comparative Analysis of Python Web Frameworks for Web Development. *Journal of Emerging Trends in Computing and Information Sciences*.
- A. Singh, M. B. (2018). A Review on Python Web Frameworks. *International Journal of Advanced Research in Computer Science and Software Engineering*.
- A. Ademola, & T. Ademola. (2019). A Comparative Analysis of Front-End Web Development Frameworks. *Journal of Computer Science and Its Applications*.
- Amazon. (2023). *Amazon Web Services*. Recuperado el 7 de Julio de 2023, de Amazon Web Services: <https://aws.amazon.com/es/what-is/javascript/>
- Amazon AWS. (2023). *¿Qué es Java?* Recuperado el 8 de Julio de 2023, de Amazon AWS: <https://aws.amazon.com/es/what-is/java/>
- Anderson, D. J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*.
- Arencibia, Y. O. (2018). *Herramienta eXcriba Migration Tool versión 2.0 para el sistema gestor de documentos administrativos XABAL EXCRIBA 3.1*.
- Baluja, W. G., & Soler, J. A. (2020). Aspectos claves para la infomatización y el Gobierno Electrónico. *Revista Cubana de Ciencias Informáticas*, 14(3).
- Beatriz Elena Cano Maya, J. J. (2014). *SOFTWARE COMO UN SERVICIO (SaaS), UN MODELO DE ACCESO A LAS TIC*.
- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*.
- Beck, K. (2019). *Extreme Programming Explained: Embrace Change*. *IEEE Access*.
- Bludid, C. H. (2019). *El SIGA y su incidencia en el control de gasto del fondo de caja chica del Seguro Integral de Salud*. Lima.
- Briseño, A. D. (15 de Marzo de 2021). *EuroSoft Software Development*. Recuperado el 2021 de Mayo de 18, de Sistema de control de comedores: <https://solucioneseurosoft.com/control-acceso/sistema-de-control-de-comedores/#:~:text=Con%20el%20sistema%20de%20control%20de%20comedores%20de,8%20Adaptabilidad%20a%20las%20pol%C3%ADticas%20de%20la%20empresa>
- Center for Democracy & Technology. (2023). *Cybersecurity*. Recuperado el 2 de Septiembre de 2023, de Center for Democracy & Technology: <https://cdt.org/area-of-focus/cybersecurity-standards/cybersecurity/>
- Chillarón, D. C. (2016). *Desarrollo de una plataforma social para compartir imágenes en dispositivos Android*. Madrid: Universidad Politécnica de Madrid.

- Claire Rowland, E. G. (2020). *Designing Connected Products: UX for the Consumer Internet of Things*.
- Comisión Europea. (2018). *Protección de datos: Normas sobre protección de datos personales dentro y fuera de la UE*. Recuperado el 2 de Septiembre de 2023, de Comisión Europea: https://commission.europa.eu/law/law-topic/data-protection_es
- Coppola, M. (2023). *HubSpot*. Recuperado el 19 de Mayo de 2023, de Frontend y backend: qué son, en qué se diferencian y ejemplos: <https://blog.hubspot.es/website/frontend-y-backend>
- Cubides, J. S. (2020). Panorama histórico de la Ciencia y la Tecnología. *Ciencia y Poder Aéreo*.
- DesarrolloWeb. (2019). *DesarrolloWeb*. Recuperado el 5 de Mayo de 2023, de Manual de CSS3: <https://desarrolloweb.com/manuales/css3.html>
- Díaz, J. G. (2016). *Módulos de visualización de la información paciente-especialista para la plataforma de gestión de videojuegos Medicando*.
- Django Rest Framework. (2022). *Django Rest Framework*. Recuperado el 5 de Mayo de 2023, de Marco de Django Rest Framework: <https://www.django-rest-framework.org/>
- Duque, R. G. (2019). *Python para todos*. Madrid: Creative Commons .
- Emilia Mendes, N. M. (2005). *Web Engineering*. Springer .
- Farro, J. (2016). *SISTEMA INTEGRADO DE GESTIÓN ADMINISTRATIVA*. Perú.
- FastAPI. (2023). *FastAPI Documentation*. Recuperado el 8 de Julio de 2023, de FastAPI: <https://fastapi.tiangolo.com/>
- Flanagan, D. (2020). *JavaScript: The Definitive Guide*.
- FLASK. (2022). *FLASK*. Recuperado el 20 de Mayo de 2023, de FLASK DOCUMENTATION: <https://flask.palletsprojects.com/en/2.3.x/>
- Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional.
- Gene Kim, J. H. (2022). *How to Create World-Class Agility, Reliability, and Security in Technology Organizations*.
- Git. (2023). *Git Documentation*. Recuperado el 9 de Julio de 2023, de Git: <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Fundamentos-de-Git>
- GitHub. (2023). *GitHub*. Recuperado el 7 de Julio de 2023, de Asistencia de GitHub: <https://support.github.com/>
- GitHub Octoverse. (2022). *GitHub Octoverse*. Recuperado el 7 de Julio de 2023, de Top programming languages: <https://octoverse.github.com/2022/top-programming-languages>

- GitHub Octoverse. (2023). *GitHub Octoverse*. Recuperado el 7 de Julio de 2023, de GitHub Octoverse: <https://octoverse.github.com/>
- Gutierrez, D. (2020). *¿Por qué el desarrollo de software personalizado es mejor para los negocios?* Recuperado el 17 de Agosto de 2023, de VELNEO: <https://www.velneo.com/blog/desarrollo-software-personalizado-mejora-negocios>
- Gutierrez, M. E. (2020). *Estudio comparativo de la implementación de una aplicación web o de escritorio para el control del personal de la empresa AMESEPREVE en la ciudad de Babahoyo*.
- Gutiérrez, R. T. (2016). *Teknokultura. Revista de Cultura Digital y Movimientos Sociales*. Recuperado el 5 de Mayo de 2023, de El surgimiento de HTML5; un nuevo paradigma en los estándares Web: <https://revistas.ucm.es/index.php/TEKN/article/view/52152>
- H. Al-Rababah, S. A.-S. (2018). Frontend and backend web development: an overview. *Journal of Physics: Conference Series*.
- Head, V. (2020). *Experience, Designing Interface Animation: Meaningful Motion for User*.
- Hernández, S. S. (2017). *Desarrollo de aplicaciones web con PHP y MySQL*. Madrid: Ra-Ma Editorial.
- Hernández-Rodríguez, N. R. (2011). Diseño de un modelo general para la gestión de sistemas logísticos en empresas cubanas: consideraciones teóricas y prácticas. *Santiago*.
- (2014). *Implementación del modelo integral colaborativo como fuente de innovación para el desarrollo ágil de software en las empresas de la zona Centro- Occidente de México*.
- Ingeaa. (2023). *Ingeaa*. Recuperado el 18 de Mayo de 2023, de Control de comedor local: <https://www.ingeeaa.com.mx/control-de-comedor>
- Ivet Challenger-Pérez, Y. D.-R.-G. (2014). El lenguaje de programación Python. *Centro de Información y Gestión Tecnológica*.
- J. C. Cubero, J. M. (2012). Gestión de bases de datos. *Revista de Investigación en Tecnologías de la Información y Comunicación*.
- Javaria Akram, S. I. (2021). Workplace food management systems for improving employee nutrition: A systematic review. *Nutrition & Food Science. Nutrition & Food Science*.
- Jiménez, J. D. (2019). *OpenWebinars*. Recuperado el 5 de Mayo de 2023, de ¿Qué es HTML5?: <https://openwebinars.net/blog/que-es-html5/>
- JQuery. (2023). *JQuery*. Recuperado el 19 de Mayo de 2023, de What is jQuery?: <https://jquery.com/>
- Kanban University. (2023). *The Official Guide to the Kanban method*. Recuperado el 3 de Septiembre de 2023, de Kanban University: <https://kanban.university/kanban-guide/>
- Kenneth C. Laudon, J. P. (2012). *Sistemas de información gerencial* (12 ed.). PEARSON .

- Kent Beck, C. A. (2004). *Extreme Programming Explained: Embrace Change (2nd ed.)*. Addison-Wesley.
- Kim, Y.-K. (2020). The effects of a food service management system on foodservice quality, patient satisfaction, and waste reduction in a university hospital. *Journal of Foodservice Business Research*.
- L. Wu, E. K. (2020). jQuery and Its Competitors: An Empirical Study of How JavaScript Developers Use Web Frameworks. *IEEE Transactions on Software Engineering*.
- Larry L. Peterson, B. S. (2019). *Computer Networks: A Systems Approach*.
- Leal, M. (27 de Abril de 2022). *Foodtech*. Recuperado el 2023 de Mayo de 3, de Gestión en un servicio de alimentación: <https://redalimentariafoodtech.com/nota/501082-gestion-en-un-servicio-de-alimentacion#:~:text=La%20GESTI%C3%93N%20DEL%20SERVICIO%20DE,y%20garantizar%20que%20las%20necesidades>
- León, B. (2014). *Sistema Web Móvil para realizar reservas de menú en el Centro de Practicas Pre- Profesionales de alimentos y bebidas El Mesón del Estudiante de la Universidad Ricardo Palma*. Universidad Ricardo Palma.
- Luicho, R. D. (2020). *Implementación del SIGA MEF y su influencia en las contrataciones de bienes y servicios en la municipalidad de Potoni*.
- Lutz, M. (2021). *Learning Python*.
- M. Alam, M. M. (2017). A Review of Backend Technologies for Web Development. *Journal of Computer and Communications*.
- Marqués, M. (2011). *Bases de datos*.
- McNeil, C. &. (2020). *CSS in Depth*. Manning Publications.
- MealSuite. (2023). *MealSuite*. Recuperado el 5 de mayo de 2023, de MealSuite: <https://www.mealsuite.com/>
- Mercado, H. E., & Eguiarte, M. d. (2019). *La incorporación de las TIC en las empresas. Factores de la brecha digital en las Mipymes de Aguascalientes*. México.
- Microsoft. (2023). *Microsoft Accessibility*. Recuperado el 2 de Septiembre de 2023, de Microsoft: <https://www.microsoft.com/en-us/accessibility?rtc=1>
- Microsoft Learn. (2023). *¿Qué es la administración de identidad y acceso?* Recuperado el 31 de Agosto de 2023, de Microsoft Learn: <https://learn.microsoft.com/es-es/azure/active-directory/fundamentals/introduction-identity-access-management>
- Microsoft Learn. (2023). *Acerca de la seguridad, la autenticación y la autorización*. Recuperado el 31 de Agosto de 2023, de Microsoft Learn: <https://learn.microsoft.com/es-es/azure/devops/organizations/security/about-security-identity?view=azure-devops>

- Microsoft Support. (2023). *Conceptos básicos del diseño de una base de datos*. Recuperado el 31 de Agosto de 2023, de Microsoft Support: <https://support.microsoft.com/es-es/office/conceptos-b%C3%A1sicos-del-dise%C3%B1o-de-una-base-de-datos-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5>
- Miguel Angel Alvarez, F. E. (2017). *DesarrolloWeb*. Recuperado el 5 de Mayo de 2023, de Manual de CSS, hojas de estilo: <https://kechy.files.wordpress.com/2012/11/manual-css-hojas-estilo.pdf>
- Ministerio de Economía y Finanzas. (2021). *Módulo de Configuración*. Perú.
- Mirza, O. (2019). Choosing the Right Technology Stack for Web Application Development. *International Journal of Advanced Computer Science and Applications*.
- Monsalve, H. S. (2007). *Diseño e implementación del sistema de gestión de comedor de la oficina general de bienestar universitario de la Universidad Nacional de la Amazonía Peruana*. Trabajo de Diploma .
- Montoro, A. F. (2012). *Python al descubierto*. Madrid: RC Libros.
- Mory, F. J. (2021). *Revista Empresarial & Laboral*. Recuperado el 19 de Mayo de 2023, de Metodologías Ágiles vs Tradicionales: <https://revistaempresarial.com/tecnologia/las-implicaciones-energeticas-de-la-5g-en-el-nucleo-de-las-telecomunicaciones/>
- Mujica, M. (2000). Nuevas estrategias para gerenciar, una visión epistemológica. *UNESR* , 1(1).
- MUÑOYA, D. P. (2022). *Agencia Cubana de Noticias*. Recuperado el 19 de Mayo de 2023, de Crea XETID sistema informático para la gestión empresarial: <http://www.acn.cu/medio-ambiente/92320-crea-xetid-sistema-informatico-para-la-gestion-empresarial>
- N. K. Sharma, V. S. (2015). An Overview of Backend Frameworks for Web Development. *International Journal of Advanced Research in Computer Science and Software Engineering*.
- Nekaka, P. M. (2020). *Sistema de Gestión de Estudiantes para el Sector de Apoyo Estudiantil de Angola en Cuba*.
- Nielsen, J. (2020). *10 Usability Heuristics for User Interface Design*. Recuperado el 2 de Septiembre de 2023, de Nielsen Norman Group: <https://www.nngroup.com/articles/ten-usability-heuristics/>
- NIST. (2011). *National Institute of Standards and Technology*. Recuperado el 10 de Mayo de 2023, de The NIST Definition of Cloud Computing: <https://csrc.nist.gov/publications/detail/sp/800-145/final>
- Node.js. (2023). *Node.js*. Recuperado el 19 de Mayo de 2023, de Acerca de Node.js: <https://nodejs.org/es/about>
- Nutrislice. (2023). *Nutrislice*. Recuperado el 3 de Mayo de 2023, de Nutrislice: <https://nutrislice.com/>

- O`Grady, S. (16 de Mayo de 2023). *Clasificación del lenguaje de programación RedMonk: enero de 2023*. Recuperado el 7 de Julio de 2023, de RedMonk: <https://redmonk.com/sogrady/2023/05/16/language-rankings-1-23/>
- Oliver, I. (2014). *Privacy Engineering: A Dataflow and Ontological Approach*.
- Olsina L., R. G. (2016). *Desarrollo de Software: Requisitos, Análisis y Diseño*. Universidad Nacional de La Plata.
- Oracle. (2022). *Configurar usuarios, roles de acceso y permisos*. Recuperado el 31 de Agosto de 2023, de Documentación de Oracle Cloud Infrastructure: <https://docs.oracle.com/es-ww/iaas/blockchain-platform/doc/set-users-and-application-roles.html>
- Oraldo Jacinto Simón, A. V. (2016). *SISTEMA DE GESTION EMPRESARIAL DISTRA Y EL ASEGURAMIENTO DE LA CALIDAD DEL PROCESO Y EL PRODUCTO*. La Habana.
- Oviedo-García M. A., G.-S. F.-P. (2013). Improving Software Quality by Using GRASP and GoF Design Patterns. *Journal of Universal Computer Science*.
- Padilla, R. (2013). *Ingeniería en sistemas*. México: Alfaomega.
- Patricia López, F. R. (2014). *INGENIERÍA DEL SOFTWARE I, Lenguaje Unificado de Modelado - UML*. Cantabria.
- Petukhova, E. (2019). *Sitecore JavaScript Services Framework Comparison*.
- PHP. (2023). *¿Qué es PHP?* Recuperado el 8 de Julio de 2023, de PHP: <https://www.php.net/manual/es/intro-what-is.php>
- Pons, O. &. (2017). Requisitos no funcionales en el desarrollo de software. *Revista de Sistemas de Información*.
- Pressman, R. S. (2010). *Ingeniería del Software: Un Enfoque Práctico*. McGraw-Hill.
- Pydantic. (2023). *Welcome to Pydantic*. Recuperado el 9 de Julio de 2023, de Pydantic: <https://docs.pydantic.dev/latest/>
- R&C Consulting. (2019). Recuperado el 19 de Mayo de 2023, de QUÉ ES EL SIGA? (SISTEMA INTEGRADO DE GESTIÓN ADMINISTRATIVA): <https://rc-consulting.org/blog/2019/06/que-es-el-siga/>
- R&C Consulting. (2019). *R&C Consulting*. Recuperado el 5 de Julio de 2023, de SIGA MEF : SISTEMA INTEGRADO DE GESTIÓN ADMINISTRATIVA NIVEL INTERMEDIO: <https://rc-consulting.org/sistema-informatico-de-gestion-administrativa-siga-mef-online-2/>
- R. Elmasri, S. N. (1994). Fundamentos de bases de datos. *IEEE Transactions on Knowledge and Data Engineering*.
- Randal E. Bryant, D. R. (2021). *Computer Systems: A Programmer's Perspective*.

- Raul Gabriel Urma, M. F. (2020). *Modern Java in Action*.
- REACT. (2023). *React*. Recuperado el 5 de Mayo de 2023, de La biblioteca para interfaces de usuario web y nativas: <https://es.react.dev/>
- REACT DOCUMENTATION. (2023). *REACT*. Recuperado el 5 de Mayo de 2023, de REACT DOCUMENTATION: <https://es.legacy.reactjs.org/>
- Real Academia Española. (2014). *Real Academia Española*. Recuperado el 2023 de Mayo de 4, de Diccionario de la lengua española: <https://dle.rae.es/>
- RedMonk. (2023). *RedMonk*. Recuperado el 7 de Julio de 2023, de RedMonk About: <https://redmonk.com/about/>
- Reyes, A. J. (2022). *Radio Rebelde*. Recuperado el 19 de Mayo de 2023, de Cuba cuenta con un software para la gestión empresarial: <https://www.radiorebelde.cu/noticia/cuba-cuenta-con-un-software-para-la-gestion-empresarial-20220324/>
- Ricardo Ruiz, B. V. (2015). *Módulo de Recomendaciones para el Sistema*. La Habana.
- Robert Aiello, L. S. (2019). *Configuration Management Best Practices: Practical Methods that Work in the Real World*.
- S. K. Goyal, R. P. (2015). Flask: A Lightweight Python Web Framework for Building Scalable and Secure Web Applications. *International Journal of Computer Applications*.
- Salazar, C.-P.-A. (2019). Extreme Programming: A Survey of Empirical Data from a Controlled Case Study. *IEEE Access*.
- Santa Fe Provincia. (2022). *Santa Fe Provincia*. Recuperado el 19 de Mayo de 2023, de Educación- Sistema de gestión escolar web: http://www.santafe.gov.ar/index.php/educacion/guia/get_tree_by_node?node_id=122273
- Schwaber, K. (2023). *About Scrum.org*. Recuperado el 3 de Septiembre de 2023, de Scrum.org: <https://www.scrum.org/about>
- Scrum Alliance. (2023). *WHAT IS SCRUM?* Recuperado el 3 de Septiembre de 2023, de Scrum Alliance: <https://www.scrumalliance.org/about-scrum>
- SIGAE WEB. (2018). *TUTORIAL DE SIGAE WEB*. Argentina.
- SIGAEWEB. (2010). *DocPlayer*. Recuperado el 5 de Julio de 2023, de TUTORIAL DE SIGAE WEB: <https://docplayer.es/1253130-Tutorial-sistema-de-gestion-escolar.html>
- SIGAEWEB. (2013). *Manual de usuario*. Argentina.
- Smith, J. P. (2018). *Entity Framework Core in Action*.
- Sommerville, I. (2011). *Ingeniería de Software*.

- Sone. (2015). *Sistema de Información de Logística para la Gestión de Insumos y Productos en una empresa del Rubro de Panadería y Pastelería*. Perú: Universidad Católica de Perú.
- Sotomayor, S. G. (2021). *Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa*. Recuperado el 3 de Septiembre de 2023, de <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
- Stackoverflow. (2022). *Stackoverflow*. Recuperado el 19 de Mayo de 2023, de 2022 Developer Survey: <https://survey.stackoverflow.co/2022/#most-popular-technologies-webframe-prof>
- Sutherland, K. S. (2020). *The Scrum Guide*.
- The PostgreSQL Global Development Group. (2023). *PostgreSQL 15.2 Documentation*. The PostgreSQL Global Development Group.
- Thomas A. Limoncelli, C. J. (2020). *The Practice of System and Network Administration*.
- Trujillo, I. L. (2007). *Propuesta de Proceso de Software para sistemas de Realidad Virtual*. La Habana.
- TuDashboard. (2022). *TuDashboard*. Recuperado el 7 de Julio de 2023, de Gráfica de dispersión. Qué es y cuáles son sus características: <https://tudashboard.com/grafica-de-dispersion/>
- Universidad de las Ciencias Informáticas. (2023). *Universidad de las Ciencias Informáticas*. Recuperado el 18 de Mayo de 2023, de La UCI de un vistazo: <https://www.uci.cu/universidad/la-uci-de-un-vistazo>
- Universidad de Santander. (2020). *Universidad de Santander*. Recuperado el 19 de Mayo de 2023, de Metodologías de desarrollo de software: ¿qué son?: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>
- Visual Studio. (2023). *Visual Studio*. Recuperado el 5 de Mayo de 2023, de Documentation for Visual Studio Code: <https://code.visualstudio.com/docs>
- W3C Web Accessibility Initiative. (2019). *Introduction to Web Accessibility*. Recuperado el 2 de Septiembre de 2023, de W3C Web Accessibility Initiative: <https://www.w3.org/WAI/fundamentals/accessibility-intro/es>
- W3Schools. (2023). *PHP Tutorial*. Recuperado el 9 de Julio de 2023, de W3Schools: <https://www.w3schools.com/php/>
- Web Accessibility in Mind. (2023). *Creating a Community that Values Accessibility*. Recuperado el 2 de Septiembre de 2023, de Web Accessibility in Mind: <https://webaim.org/articles/communities/creating>
- Weiss, M. A. (2012). *Programación en Java: Una introducción al lenguaje y sus aplicaciones*. Pearson.

Westby, E. J. (2015). *Git for Teams: A User-Centered Approach to Creating Efficient Workflows in Git*.

XETID. (2023). XETID. Recuperado el 19 de Mayo de 2023, de Gestiona y controla fácilmente tu empresa: <https://www.xetid.cu/es/productos/distra>

Zalewski, M. (2011). *The Tangled Web: A Guide to Securing Modern Web Applications*.

Zandstra, M. (2021). *PHP 8 Objects, Patterns, and Practice*.

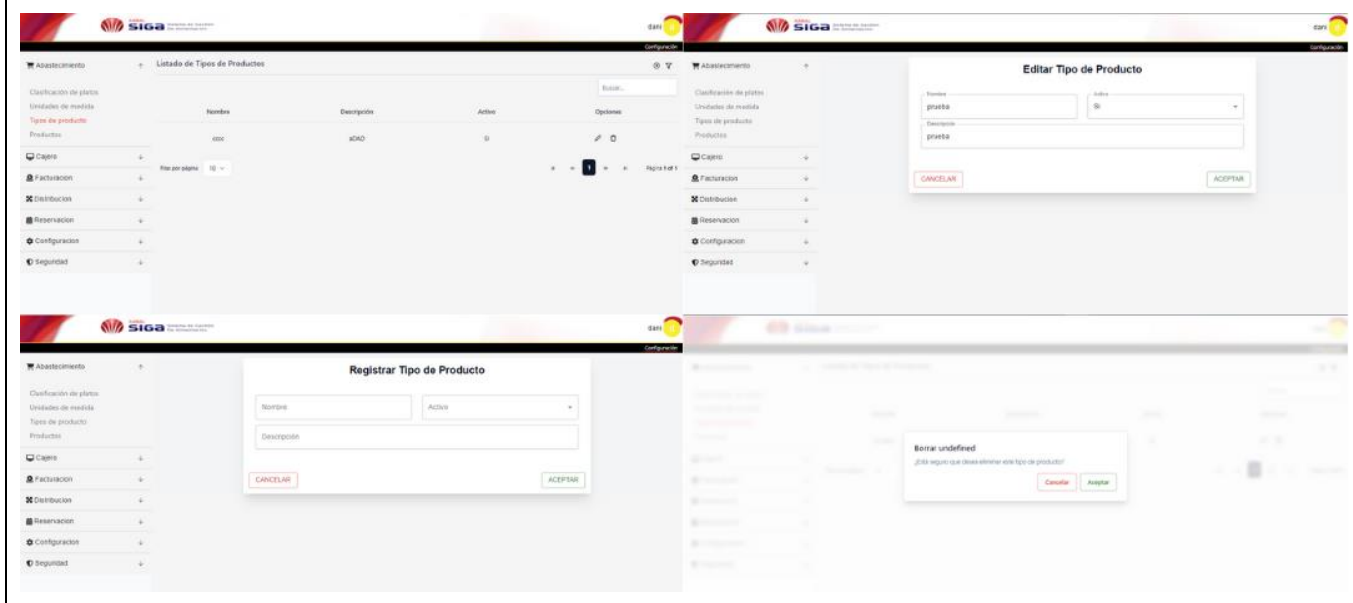
ZAPROO. (2022). ZAPROO. Recuperado el 19 de Mayo de 2023, de Enhancing User Experience: Customization vs Personalization: <https://www.zaproo.com/articles/user-experience-customization-vs-personalization>

ANEXOS

Anexo 1. Historias de usuario del módulo de configuración del sistema XABAL SIGA versión 2.0.

Historia de usuario	
Número: 3	Nombre: Gestionar tipos de productos
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración asignada: 1
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
<p>Descripción:</p> <ul style="list-style-type: none"> • Muestra una lista de tipos de producto, con el nombre, descripción, si está activo y las opciones de Registrar, Editar y Eliminar tipos de producto. Además de la posibilidad de buscar y filtrar por activo y nombre. • Permite cambiar el paginado y la cantidad de registros por página. • Registrar tipo de producto: se muestra una interfaz con los campos: nombre, activo y descripción, acompañado de los botones aceptar y cancelar. • Editar tipo de producto: se muestra una interfaz con los campos: nombre, activo y descripción, acompañado de los botones aceptar y cancelar. • Eliminar tipo de producto: se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	

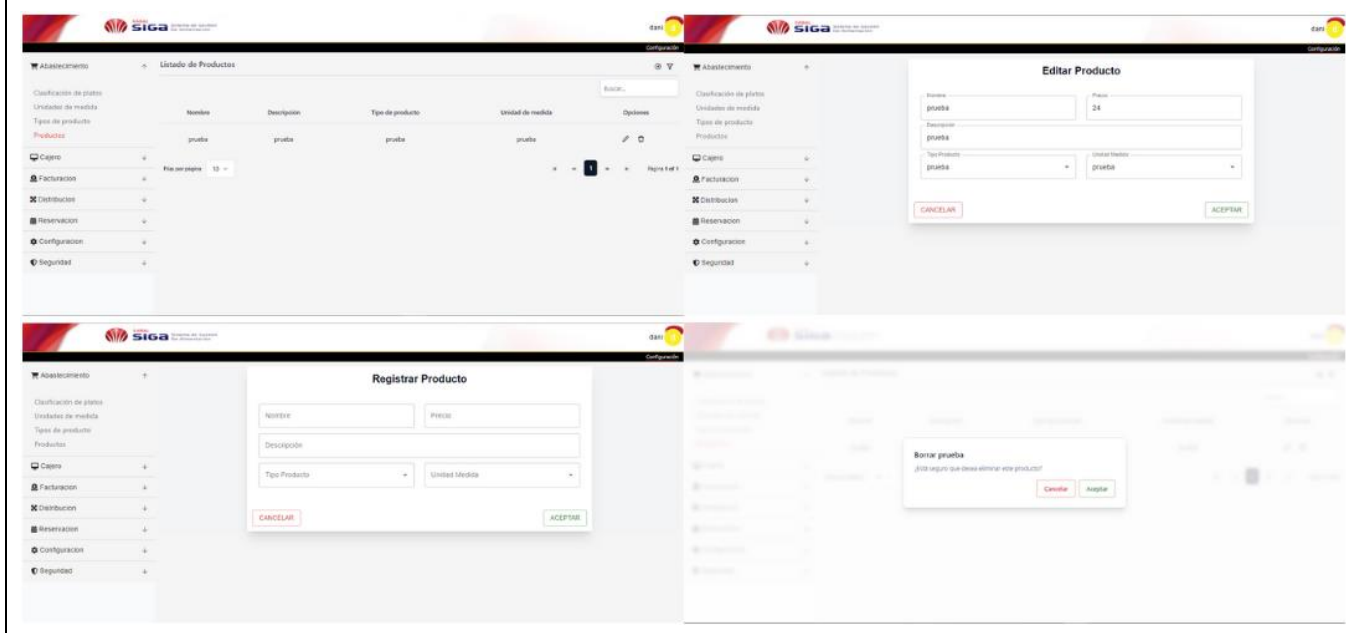
Prototipo de interfaz:



Historia de usuario	
Número: 4	Nombre: Gestionar productos
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración asignada: 1
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
<p>Descripción:</p> <ul style="list-style-type: none"> • Muestra una lista de productos, con el nombre, unidad de medida, descripción, tipo de producto y opciones de registrar, editar y eliminar. Además de la posibilidad de buscar y filtrar por nombre, tipo de producto y unidad de medida. • También permite cambiar el paginado y la cantidad de registros por página. • Registrar producto: se muestra una interfaz con los campos: nombre, unidad de medida, precio, descripción y tipo de producto, acompañado de los botones aceptar y cancelar. • Editar producto: se muestra una interfaz con los campos: nombre, unidad de medida, precio, descripción y tipo de producto, acompañado de los botones aceptar y cancelar. • Eliminar producto: se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar. 	

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



Historia de usuario

Número: 5

Nombre: Gestionar tipos de tarjetas

Usuario: Administrador general, Administrador institucional

Prioridad en negocio: Alto

Riesgo en desarrollo: Alto

Puntos estimados: 0.4

Iteración asignada: 1

Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

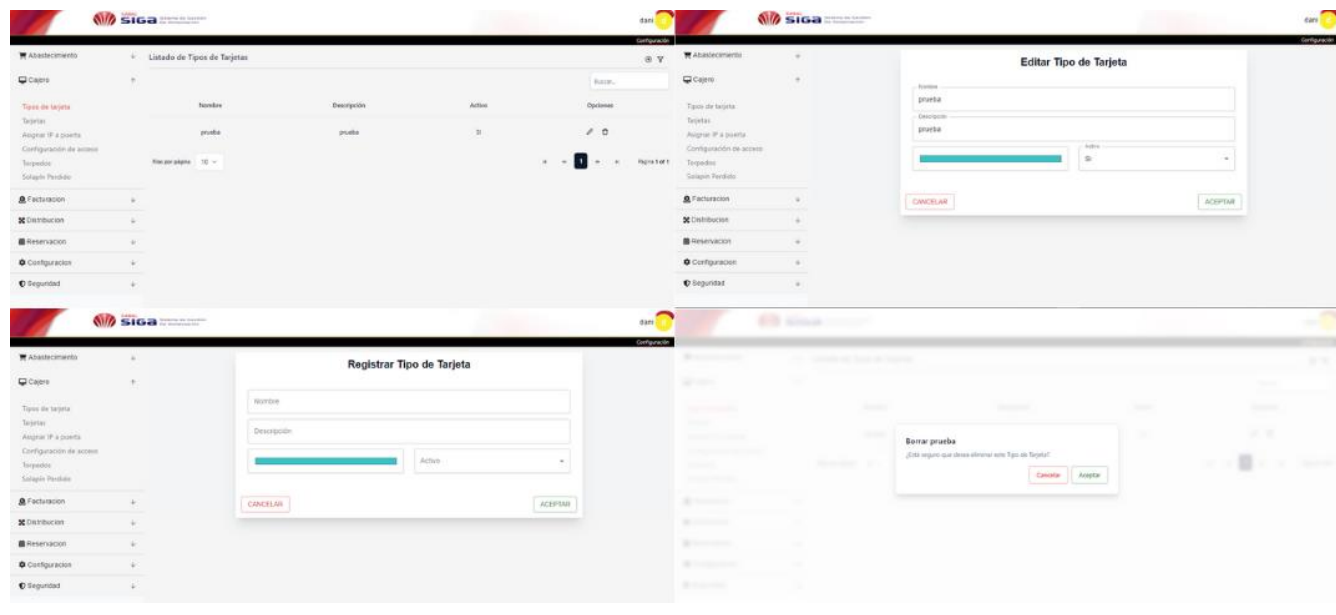
Descripción:

- Muestra una lista de tipos de tarjeta, con el nombre, descripción, si está activo y las opciones de Registrar, Editar y Eliminar tipos de tarjeta. Además de la posibilidad de buscar y filtrar por activo y nombre.
- También permite cambiar el paginado y la cantidad de registros por página.
- Registrar tipo de tarjeta: se muestra una interfaz con los campos: nombre, color, activo y descripción, acompañado de los botones aceptar y cancelar.
- Editar tipo de producto: se muestra una interfaz con los campos: nombre, color, activo y descripción, acompañado de los botones aceptar y cancelar.
- Eliminar se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que

desea eliminar? Y los botones aceptar y cancelar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



Historia de usuario

Número: 6

Nombre: Gestionar tarjetas

Usuario: Administrador general, Administrador institucional

Prioridad en negocio: Alto

Riesgo en desarrollo: Alto

Puntos estimados: 0.4

Iteración asignada: 1

Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

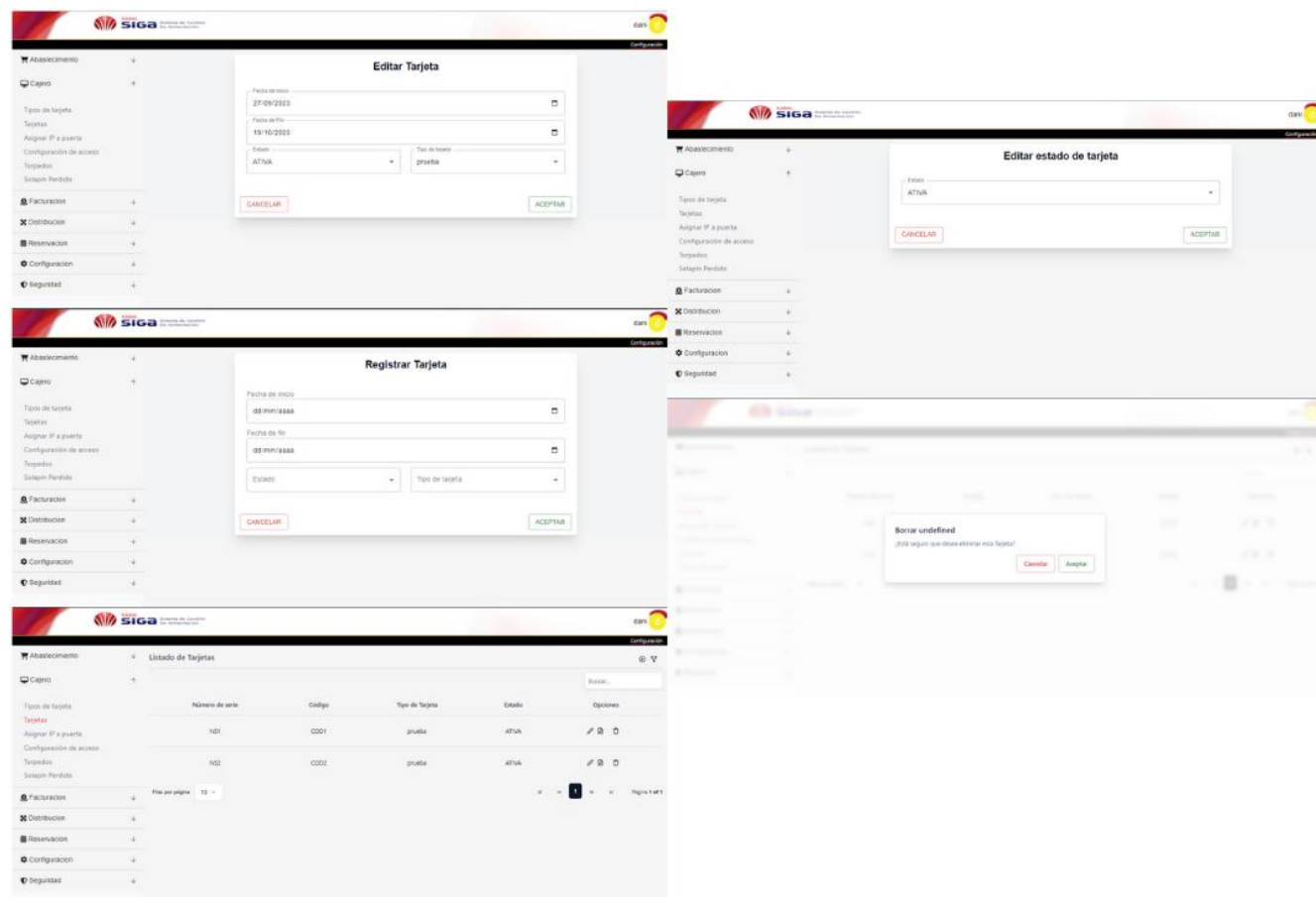
Descripción:

- Muestra una lista de tarjetas, con el número de serie, código, tipo de tarjeta, estado y las opciones de Registrar, Editar , Modificar estado, Asociar tarjeta a persona. Además de la posibilidad de buscar y filtrar por activo.
- También permite cambiar el paginado y la cantidad de registros por página.
- Registrar tarjeta: se muestra una interfaz con los campos: fecha de inicio y fin, tipo de tarjeta y estado de tarjeta, acompañado del botón aceptar y cancelar.
- Editar tarjeta: se muestra los fecha de inicio y fin, tipo de tarjeta y estado de tarjeta, acompañado del botón aceptar y cancelar.

- Al presionar modificar estado se muestra el campo estado seguido por los botones aceptar y cancelar.
- Al presionar Eliminar se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



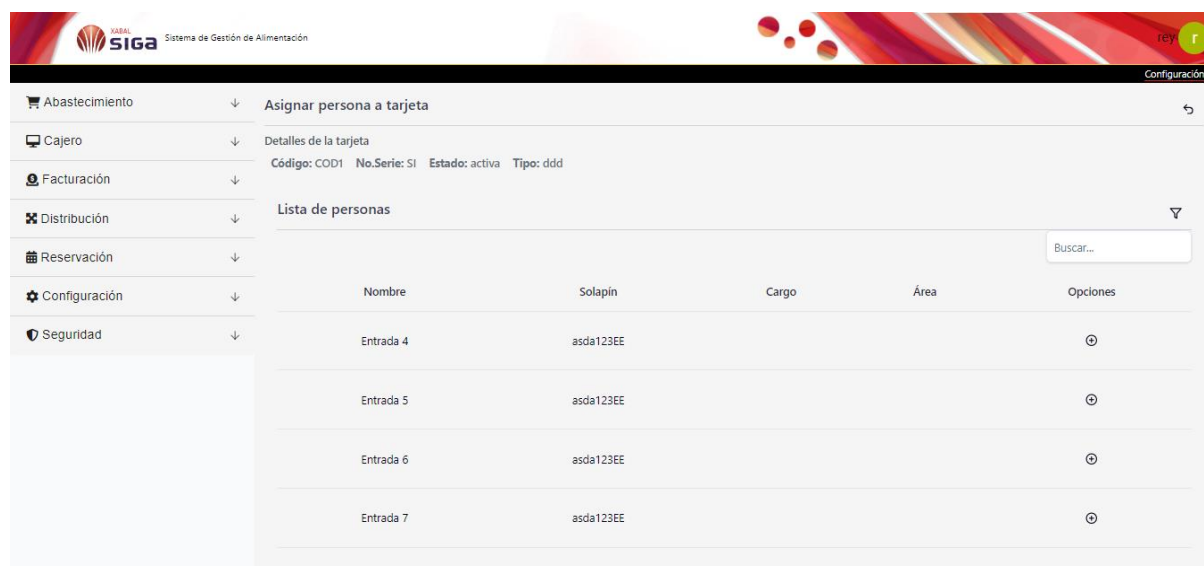
Historia de usuario	
Número: 7	Nombre: Asociar tarjeta a usuario
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 2
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	

Descripción:

- Muestra una lista de tarjetas, con el número de serie, código, tipo de tarjeta, estado y las opciones de Registrar, Ver detalles, Editar , Modificar estado y Asociar tarjeta a persona. Además de la posibilidad de buscar y filtrar por activo.
- También permite cambiar el paginado y la cantidad de registros por página.
- Al presionar asociar tarjeta a persona se muestran los detalles de la tarjeta un buscador, con un filtro por provincia, área, residente, municipio, cargo, sexo y categoría seguido por una lista de personas y el botón seleccionar y los botones aceptar y cancelar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



Historia de usuario

Número: 8

Nombre: Gestionar asignación de IP a puerta

Usuario: Administrador general, Administrador institucional

Prioridad en negocio: Alto

Riesgo en desarrollo: Alto

Puntos estimados: 0.5

Iteración asignada: 1

Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

Descripción:

- Muestra una lista de puertas activas, con el nombre, capacidad, siglas y la opción Asociar. Además de la posibilidad de buscar. También permite cambiar el paginado y la cantidad de registros

por página.

- Al presionar asociar muestran los detalles de la puerta un campo para escribir la dirección IP seguido de una lista de IP y el botón Asociar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



Historia de usuario

Número: 9 Nombre: Configurar cantidad de accesos

Usuario: Administrador general, Administrador institucional

Prioridad en negocio: Medio Riesgo en desarrollo: Medio

Puntos estimados: 0.4 Iteración asignada: 2

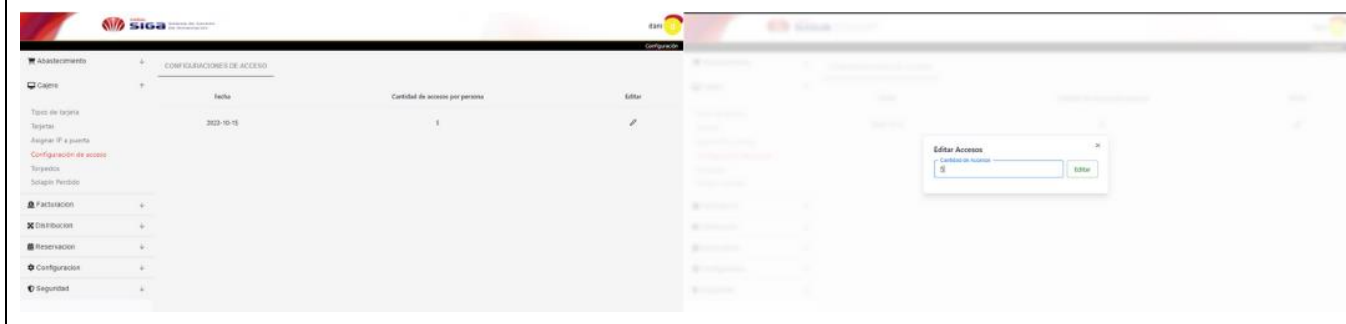
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

Descripción:

- Muestra una lista de configuraciones, con la fecha, la cantidad de accesos por persona y el botón editar.
- Editar: Se muestra un campo para definir la cantidad de accesos y el botón editar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



Historia de usuario

Número: 10

Nombre: Gestionar torpedos

Usuario: Administrador general, Administrador institucional

Prioridad en negocio: Alto

Riesgo en desarrollo: Alto

Puntos estimados: 0.5

Iteración asignada: 1

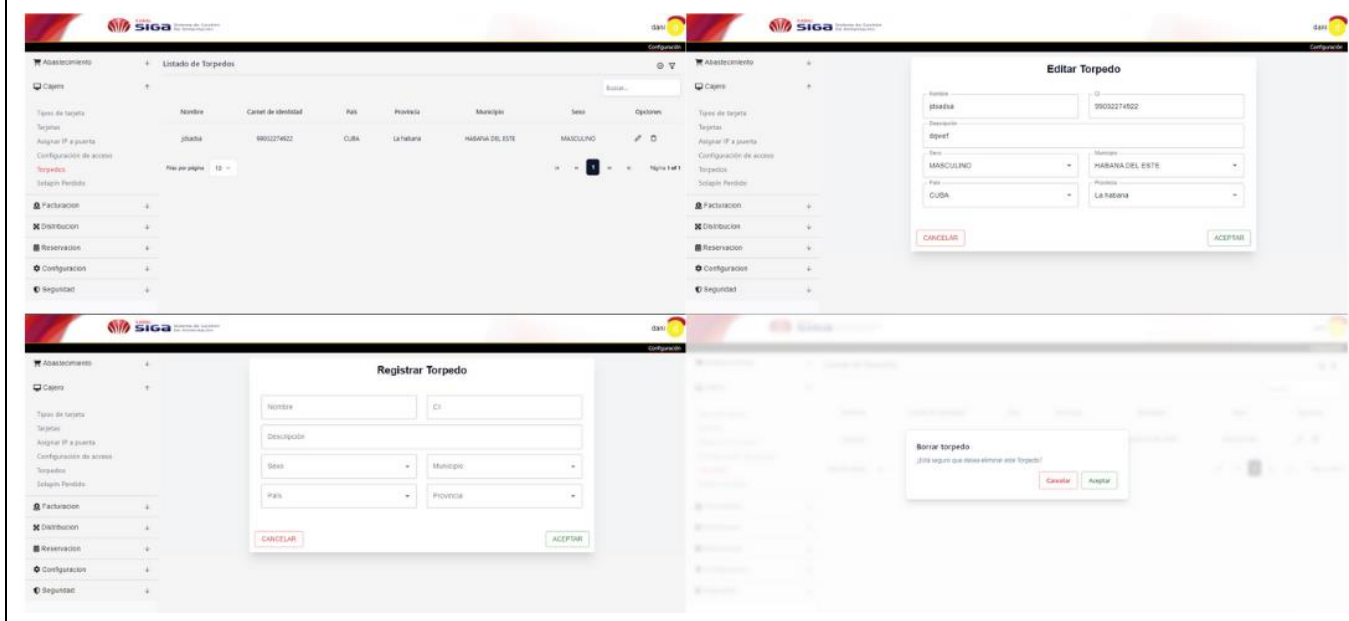
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

Descripción:

- Muestra una lista de torpedos, con el nombre, carnet de identidad, país, provincia, municipio, sexo y las opciones de registrar, editar y eliminar. Además de la posibilidad de buscar y filtrar por país, provincia, municipio, nombre, CI y sexo.
- También permite cambiar el paginado y la cantidad de registros por página.
- Registrar torpedo: muestra una interfaz con los siguientes campos: nombre, carnet de identidad, sexo, municipio, provincia, país, descripción y los botones de aceptar y cancelar.
- Eliminar: muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar.
- Editar torpedo: muestra una interfaz con los siguientes campos: nombre, carnet de identidad, sexo, municipio, provincia, país, descripción y los botones de aceptar y cancelar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



Historia de usuario

Número: 11

Nombre: Administrar solapín perdido

Usuario: Administrador general, Administrador institucional

Prioridad en negocio: Medio

Riesgo en desarrollo: Medio

Puntos estimados: 0.4

Iteración asignada: 2

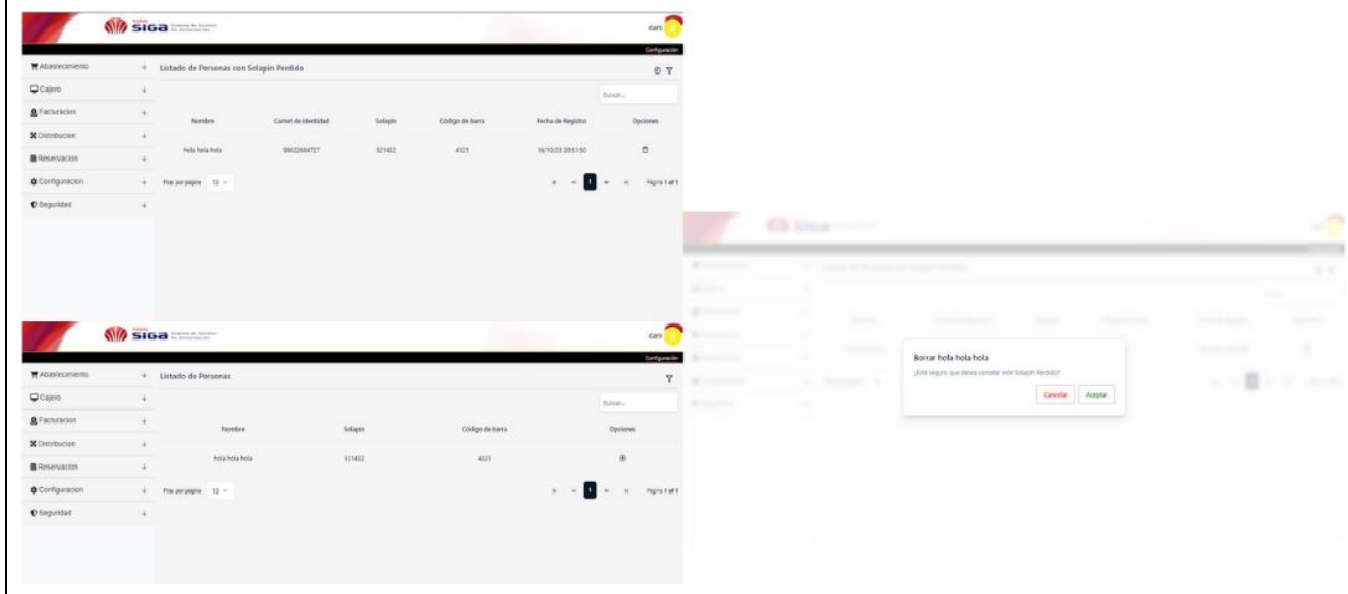
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

Descripción:

- Muestra una lista de personas con solapín perdido, con el nombre, CI, solapín, código de barra, fecha de registro y las opciones de registrar y eliminar. Además de la posibilidad de buscar y filtrar por área, cargo, categoría, residente, provincia, municipio y sexo.
- También permite cambiar el paginado y la cantidad de registros por página.
- Registrar solapín perdido: se muestra un buscador, con un filtro por provincia, área, residente, municipio, cargo, sexo y categoría seguido por una lista de personas y el botón seleccionar y el botón aceptar.
- Eliminar: se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar.

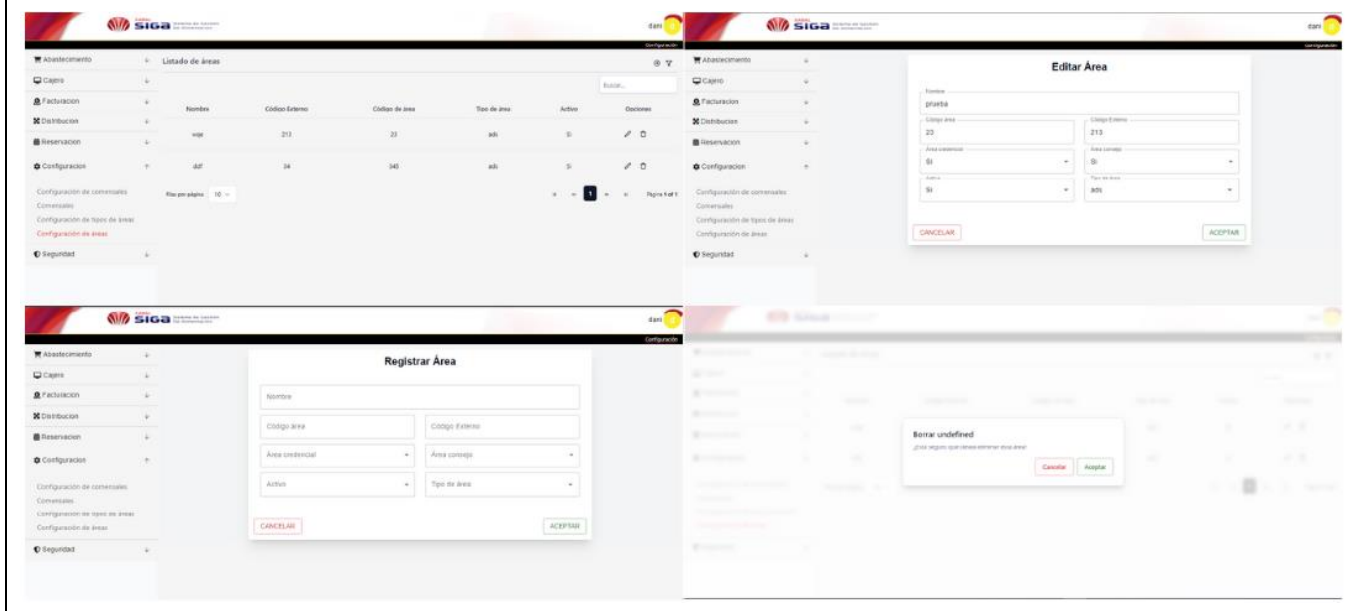
Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



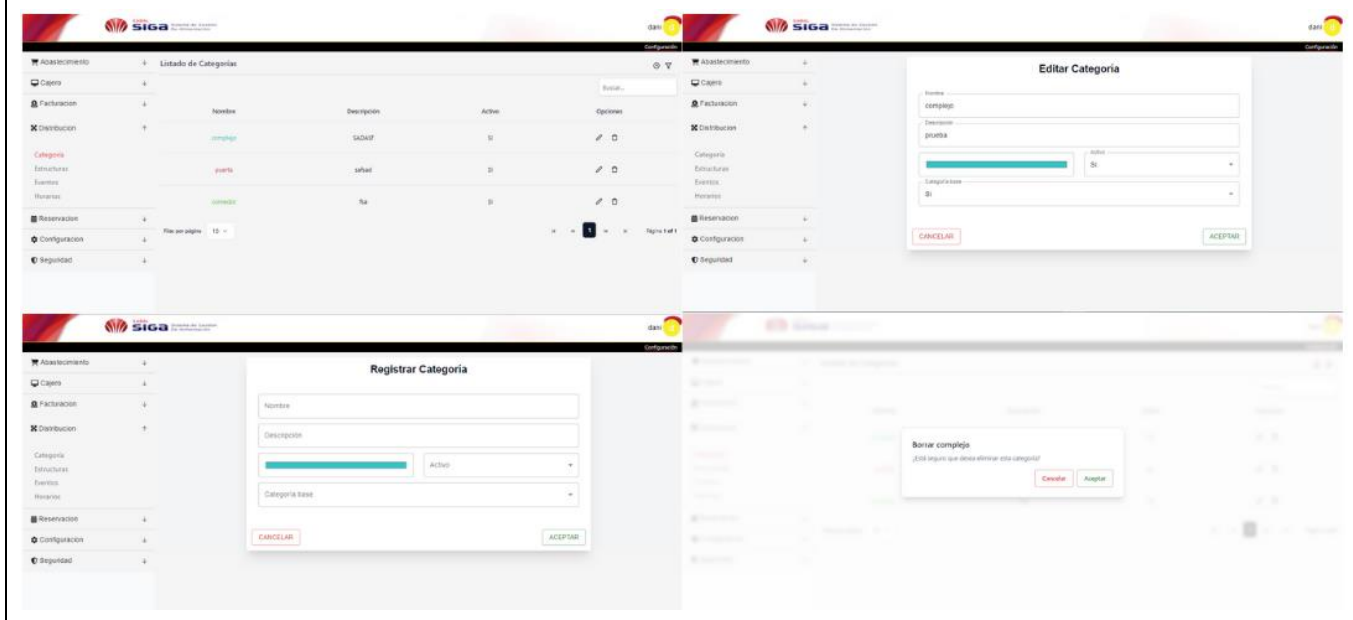
Historia de usuario	
Número: 12	Nombre: Gestionar áreas
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración asignada: 1
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
<p>Descripción:</p> <ul style="list-style-type: none"> Muestra una lista de áreas con los siguientes campos: nombre, código externo, código de área, tipo de área, activo y las opciones de registrar, editar y eliminar. Permite buscar y filtrar la búsqueda por nombre, activo y tipo de estructura. Al presionar registrar área muestra una ventana con los campos: nombre, código área, código externo, área credencial, área consejo, activo, tipo de área y los botones aceptar y cancelar. Al presionar editar área muestra una ventana con los campos: nombre, código área, código externo, área credencial, área consejo, activo, tipo de área y los botones aceptar y cancelar. Al presionar Eliminar se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	

Prototipo de interfaz:



Historia de usuario	
Número: 13	Nombre: Gestionar categorías
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración asignada: 1
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
<p>Descripción:</p> <ul style="list-style-type: none"> Muestra una lista de categorías con el nombre, descripción, activo y las opciones de registrar, editar y eliminar. Permite buscar y filtrar por activo y nombre. También permite cambiar el paginado y la cantidad de registros por página. Registrar: se muestran los campos nombre, color, activo, descripción y los botones aceptar y cancelar. Editar se muestra una interfaz con los siguientes campos nombre, color, activo, descripción y los botones aceptar y cancelar. Eliminar se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar esta área? Y los botones aceptar y cancelar. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	

Prototipo de interfaz:

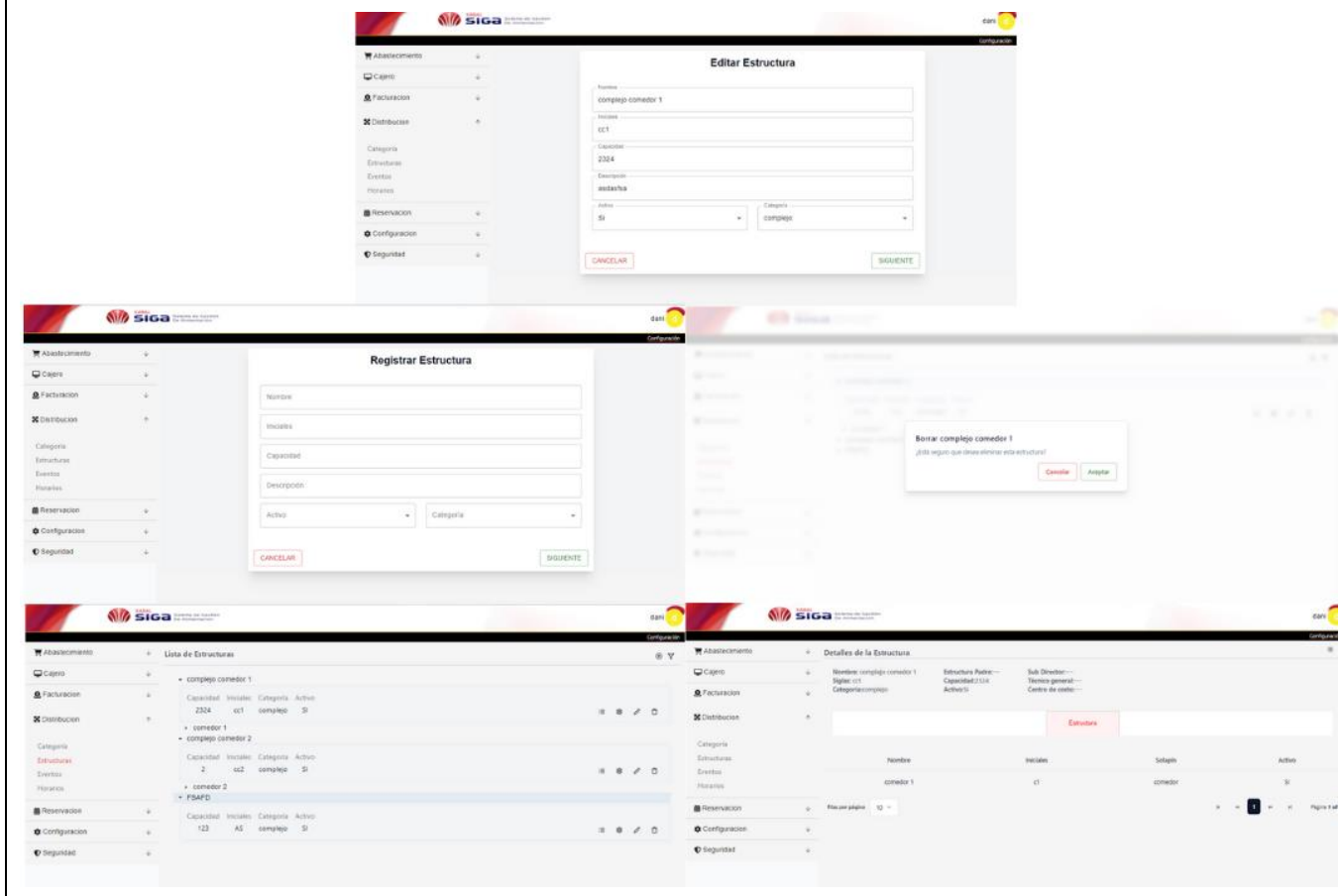


Historia de usuario	
Número: 14	Nombre: Gestionar estructuras
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración asignada: 1
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
<p>Descripción:</p> <ul style="list-style-type: none"> Muestra una vista de estructuras en forma de árbol con el nombre, capacidad, iniciales, categoría, activo y las opciones de registrar estructura, ver detalles, configurar estructura, editar y eliminar. Permite buscar y filtrar por activo, nombre e iniciales. Registrar estructura: se muestran los campos nombre, iniciales, capacidad, categoría, estructura padre, activo, descripción, reglas, valor de la regla seleccionada, acompañado de una lista de reglas y los botones asociar, aceptar y cancelar. Ver detalles: se muestra una interfaz con los datos de la estructura y a continuación las listas de estructura y responsables. Editar estructura se muestran los campos nombre, siglas, capacidad, categoría, estructura padre, activo, descripción, reglas, valor de la regla seleccionada, acompañado de una lista de reglas y los botones asociar, aceptar y cancelar. 	

- Eliminar: se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:

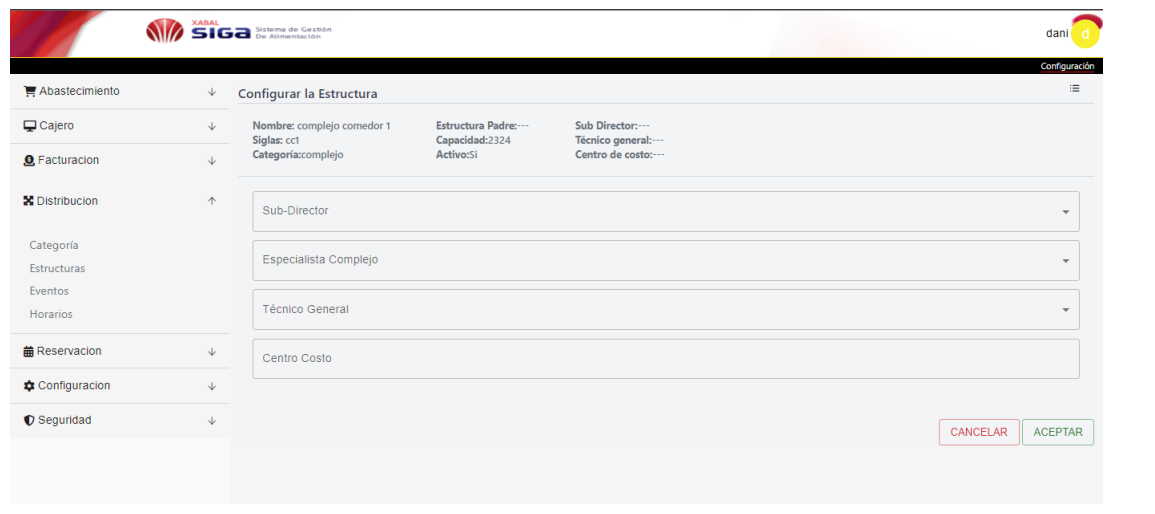


Historia de usuario	
Número: 15	Nombre: Configurar estructura
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 0.4	Iteración asignada: 2
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
Descripción:	
<ul style="list-style-type: none"> • Muestra los detalles de la estructura con el nombre, capacidad, siglas, 	

categoría, activo seguido por los campos de subdirector, técnicos generales, especialista del complejo, centro de costo y los botones aceptar y cancelar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



Historia de usuario

Número: 16

Nombre: Gestionar eventos

Usuario: Administrador general, Administrador institucional

Prioridad en negocio: Alto

Riesgo en desarrollo: Alto

Puntos estimados: 0.4

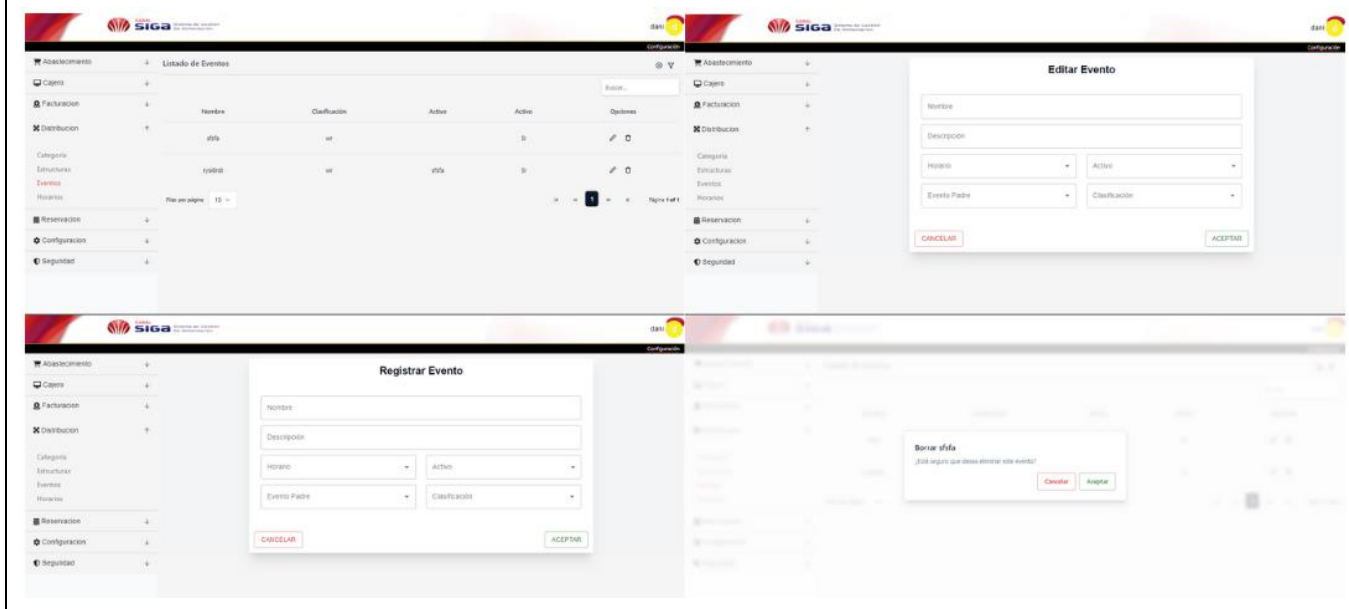
Iteración asignada: 1

Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

Descripción:

- Muestra una lista de eventos con el nombre, clasificación, evento padre, activo y las opciones de Registrar evento, Editar y Eliminar.
- También permite cambiar el paginado y la cantidad de registros por página.
- Registrar evento: se muestran los campos nombre, clasificación, evento padre, horario, activo y descripción, acompañado por los botones aceptar y cancelar.
- Editar evento: se muestran los campos nombre, clasificación, evento padre, horario, activo y descripción, acompañado por los botones aceptar y cancelar.
- Eliminar se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar.

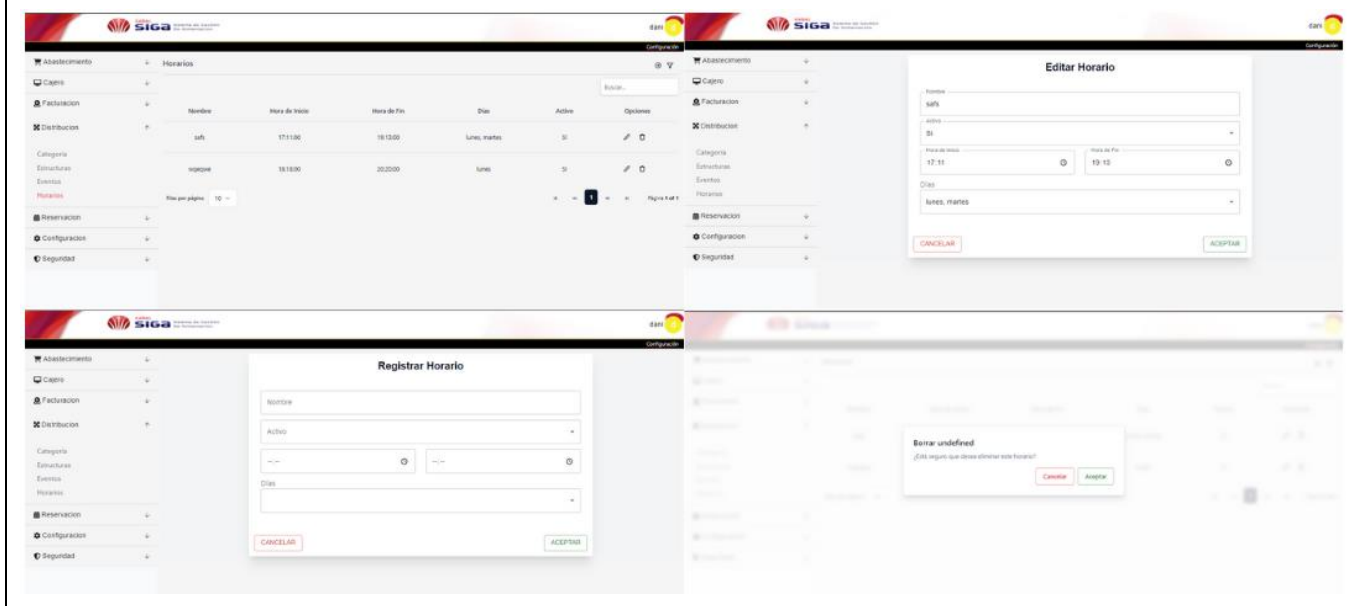
Prototipo de interfaz:



Historia de usuario	
Número: 17	Nombre: Gestionar horarios
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 0.4	Iteración asignada: 1
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
Descripción: <ul style="list-style-type: none"> Muestra una lista de horarios con el nombre, hora de inicio y fin, días de la semana, activo y las opciones de Registrar, Editar y Eliminar. También permite cambiar el paginado y la cantidad de registros por página. Permite buscar y filtrar la búsqueda por nombre y activo. Registrar horario: se muestran los campos nombre, hora de inicio y fin, días de la semana y activo, acompañado por los botones aceptar y cancelar. Editar horario: se muestran los campos nombre, hora de inicio y fin, días de la semana y activo, acompañado por los botones aceptar y cancelar. Eliminar se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar. 	

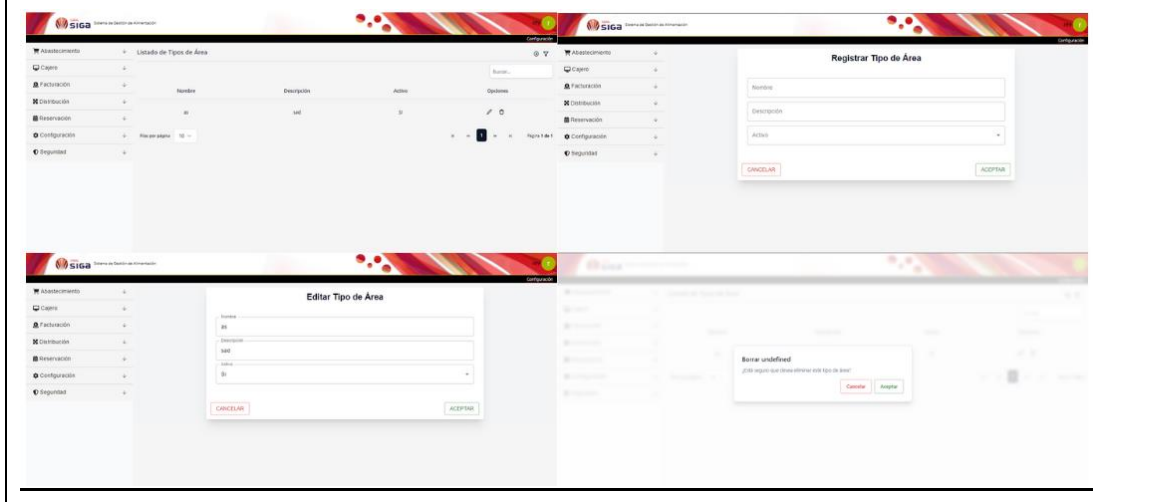
Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



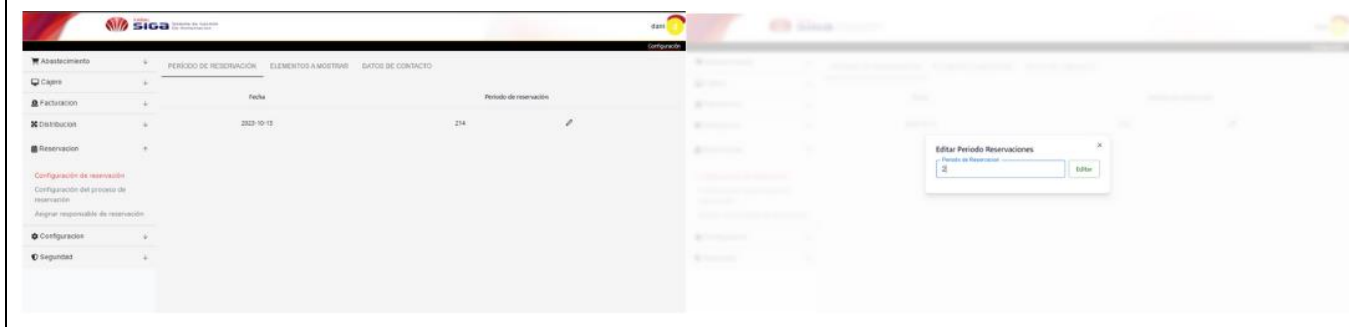
Historia de usuario	
Número: 18	Nombre: Gestionar tipos de áreas
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 0.3	Iteración asignada: 1
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
<p>Descripción:</p> <ul style="list-style-type: none"> Muestra una lista de tipos de áreas con los campos nombre, descripción, activo y las opciones registrar, editar y eliminar. Permite buscar y filtrar por nombre y activo. Registrar tipo de área: se muestra una ventana con los campos nombre, descripción y activo, seguido por los botones aceptar y cancelar. Editar tipo de área: se muestra una ventana con los campos nombre, descripción y activo, seguido por los botones aceptar. Eliminar se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	

Prototipo de interfaz:



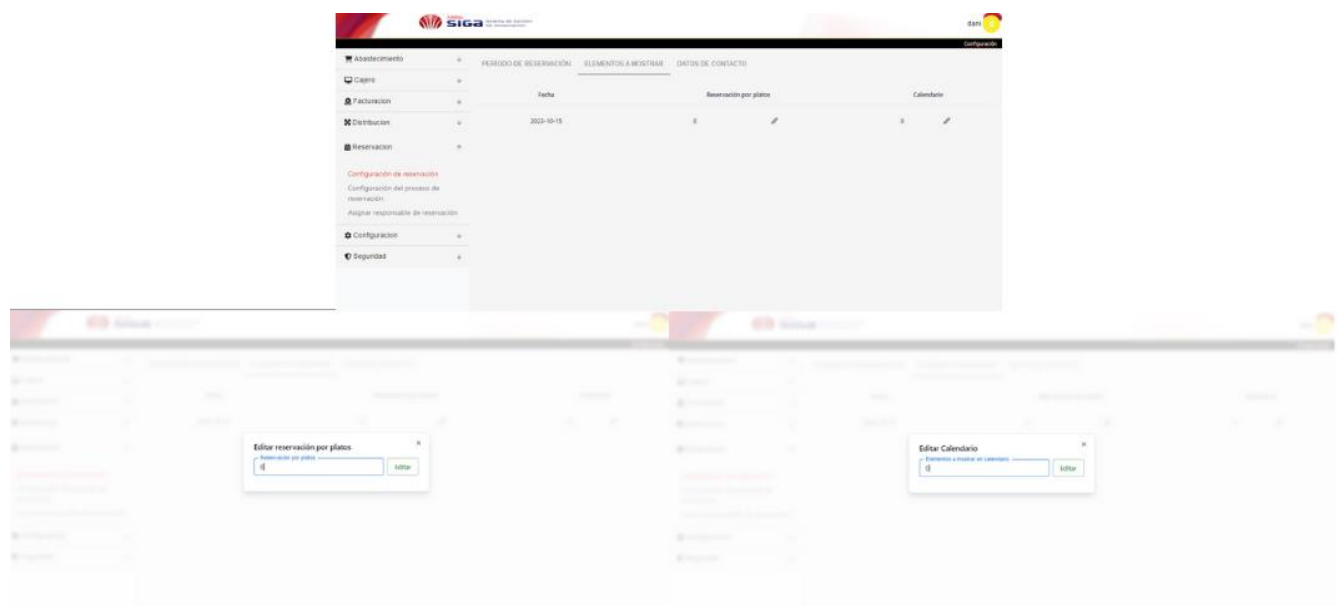
Historia de usuario	
Número: 19	Nombre: Administrar períodos de reservación
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 2
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
Descripción: <ul style="list-style-type: none"> • Muestra una lista de períodos de reservación, con la fecha y botón editar. • Editar: se muestra una interfaz con un campo para definir los períodos de reservación y el botón editar. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	

Prototipo de interfaz:



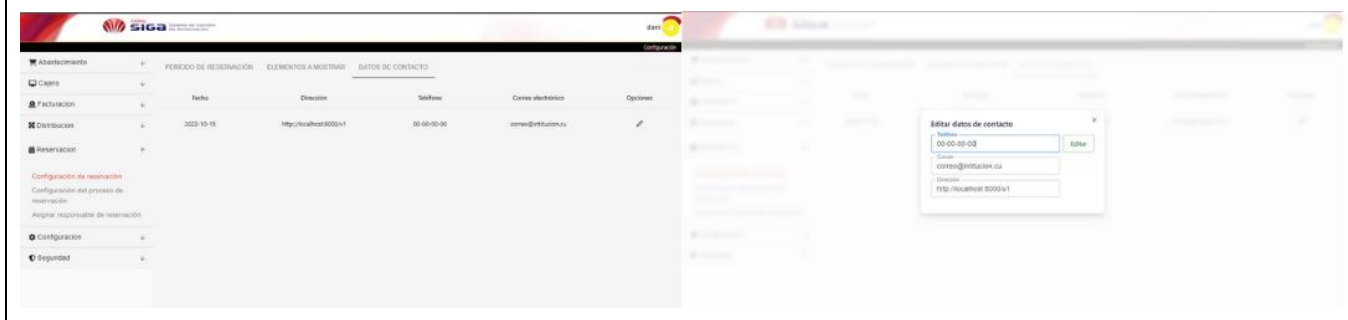
Historia de usuario	
Número: 20	Nombre: Administrar elementos a mostrar
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 2
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
Descripción: <ul style="list-style-type: none"> Muestra interfaz con la fecha, la reservación por platos y candelario y los botones editar. Editar calendario: se muestra una interfaz con un campo para definir los elementos a mostrar en el calendario y el botón editar. Editar reservación por platos: se muestra una interfaz con un campo para definir la reservación y el botón editar. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	

Prototipo de interfaz:



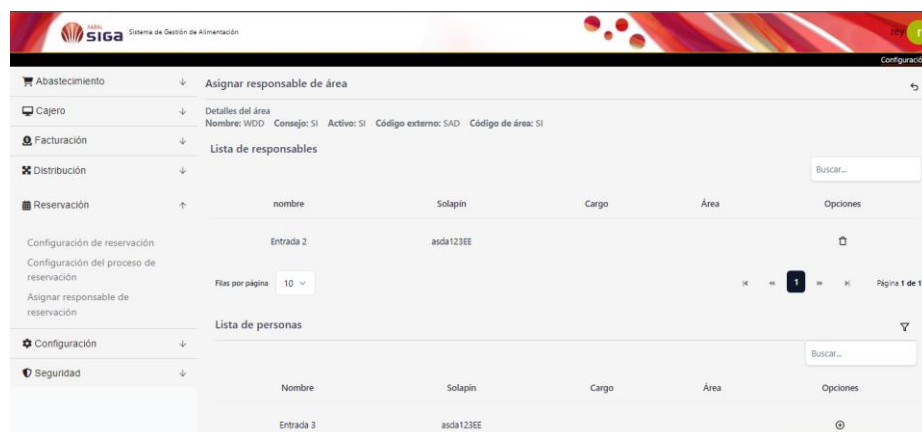
Historia de usuario	
Número: 21	Nombre: Administrar datos de contacto
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 0.4	Iteración asignada: 2
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
Descripción: <ul style="list-style-type: none"> • Muestra una lista de datos de contacto, con campo para la fecha, dirección, teléfono, correo y el botón editar. • Editar: se muestra una interfaz con los campos con campo para la fecha, dirección, teléfono, correo y el botón editar. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	

Prototipo de interfaz:



Historia de usuario	
Número: 22	Nombre: Administrar responsables de áreas
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 0.5	Iteración asignada: 2
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
Descripción: <ul style="list-style-type: none"> Muestra los detalles del área, seguido por una lista de responsables asignados al área con el botón eliminar y una lista de personas con el botón asignar. Al presionar Eliminar pasa de la lista de responsables a la lista de personas. Al presionar asignar pasa de la lista de personas a la lista de responsables. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	

Prototipo de interfaz:



Historia de usuario

Número: 23

Nombre: Gestionar módulos por institución

Usuario: Administrador general

Prioridad en negocio: Alto

Riesgo en desarrollo: Alto

Puntos estimados: 0.5

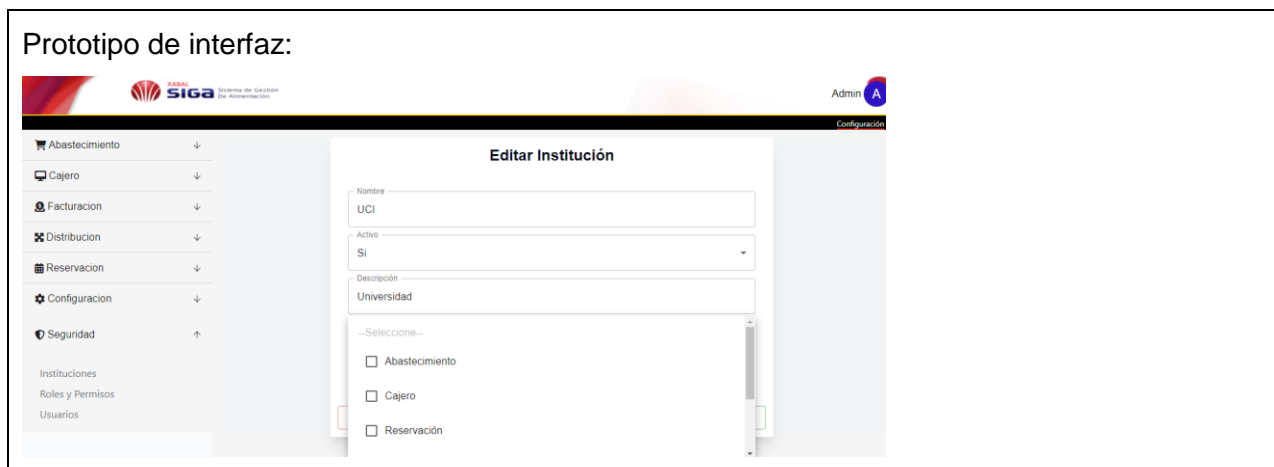
Iteración asignada: 1

Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

Descripción:

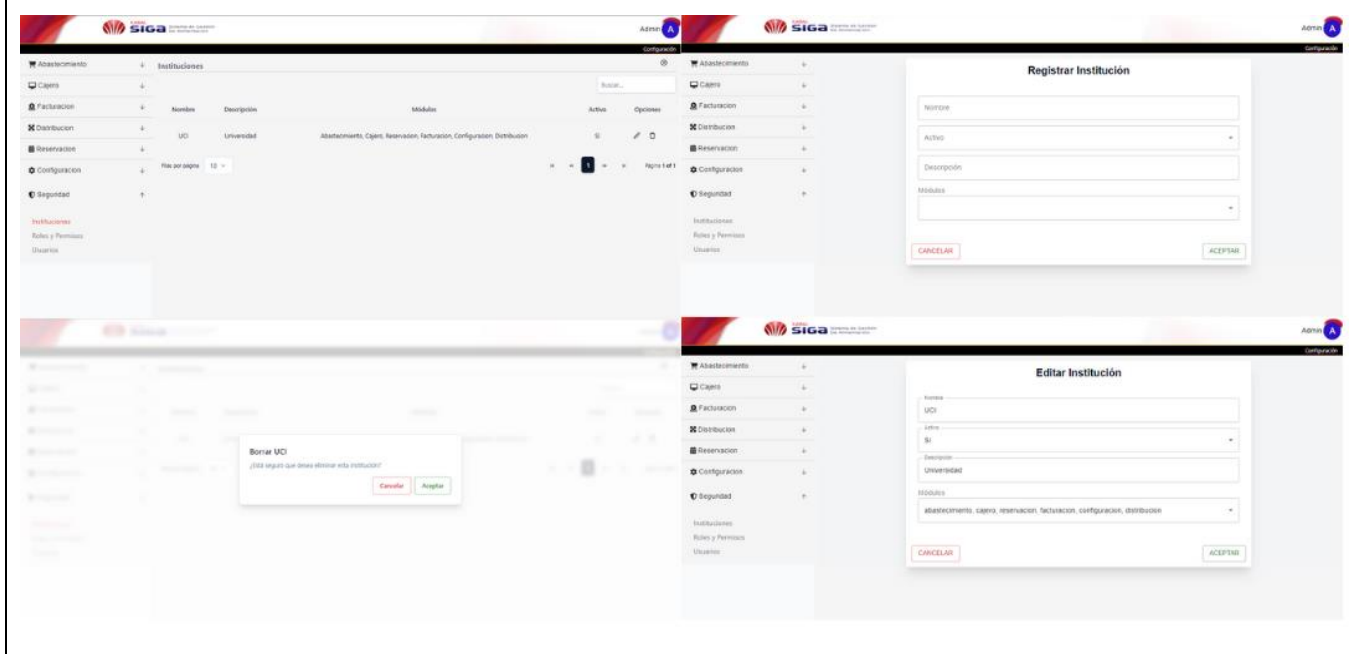
- Al presionar el botón editar institución muestra un campo para la selección de los módulos que desea que compongan su aplicación y los botones aceptar y cancelar

Observaciones: El usuario debe estar autenticado para realizar la acción.

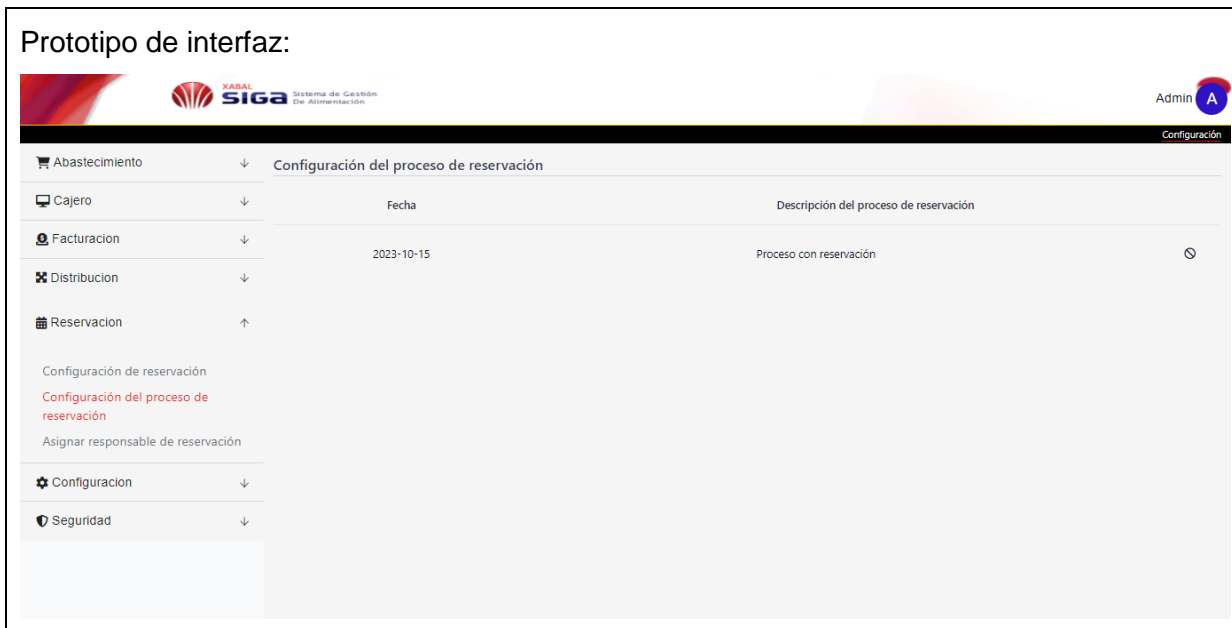


Historia de usuario	
Número: 24	Nombre: Gestionar instituciones
Usuario: Administrador general	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 0.4	Iteración asignada: 1
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
<p>Descripción:</p> <ul style="list-style-type: none"> Muestra una lista de instituciones, con nombre, descripción, módulos, activo y las opciones Registrar, Editar y Eliminar. Registrar institución se muestran los campos nombre, activo, descripción, y un campo para la selección de los módulos seguido por los botones aceptar y cancelar. Editar se muestran los campos nombre, activo, descripción, y un campo para la selección de los módulos seguido por los botones aceptar y cancelar. Al presionar Eliminar se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	

Prototipo de interfaz:

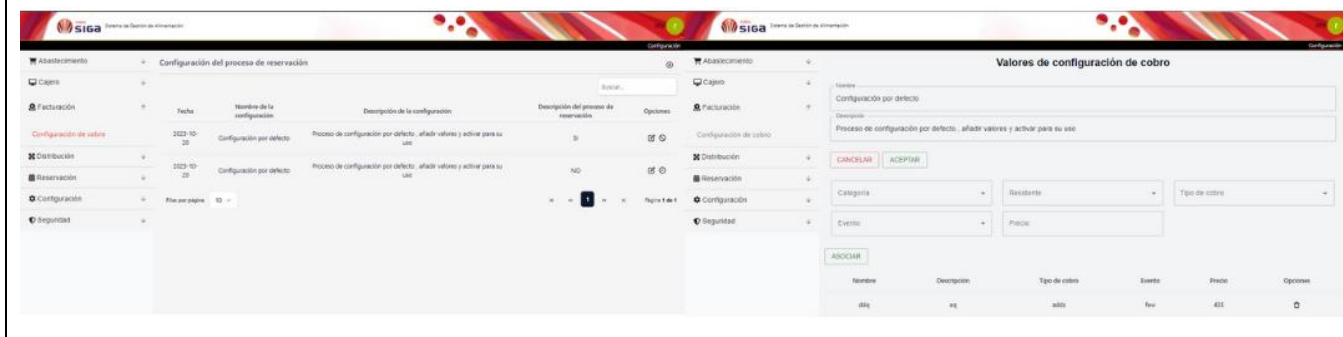


Historia de usuario	
Número: 25	Nombre: Administrar configuraciones del proceso
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 0.6	Iteración asignada: 2
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
Descripción: <ul style="list-style-type: none"> Muestra una lista de configuraciones, con la fecha, descripción, activo y un botón Modificar. Modificar se activa o desactiva el proceso con reservación. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	



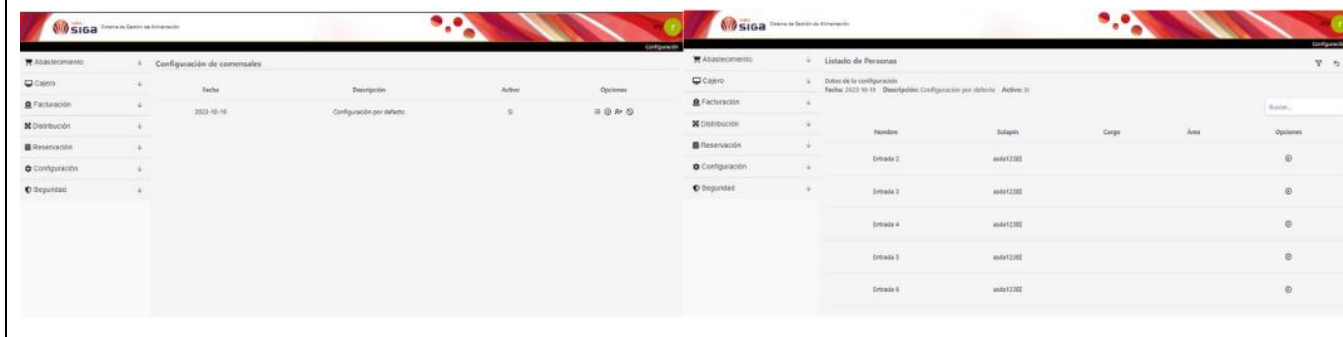
Historia de usuario	
Número: 26	Nombre: Gestionar configuración de cobro
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 0.4	Iteración asignada: 1
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
<p>Descripción:</p> <ul style="list-style-type: none"> Muestra una lista de configuraciones de cobro, con la fecha, nombre de la configuración, descripción de configuración, descripción del proceso y los botones asignar valor a configuración y activar-desactivar configuración. Al presionar el botón agregar valores a la configuración se muestra una interfaz con los campos para modificar nombre y descripción y los botones aceptar y cancelar. seguido por los campos categoría, evento, residente, precio, tipo de cobro y el botón asociar. Por último un listado con los valores que han sido asociados a esta configuración y el botón eliminar. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	

Prototipo de interfaz:



Historia de usuario	
Número: 27	Nombre: Administrar configuración de comensales
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 0.6	Iteración asignada: 2
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
<p>Descripción:</p> <ul style="list-style-type: none"> Muestra una lista de configuraciones de comensales, con la fecha, descripción, activo y las opciones Añadir valores, Adicionar comensales y desactivar-activar. Al presionar agregar valores se muestra una interfaz con los campos descripción, categoría, residente y áreas con el botón asociar y un listado de valores asignados. Al presionar asignar personas se muestra una lista de personas con el botón asignar. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	

Prototipo de interfaz:



Historia de usuario	
Número: 28	Nombre: Administrar comensales
Usuario: Administrador general, Administrador institucional	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 0.5	Iteración asignada: 2
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila	
<p>Descripción:</p> <ul style="list-style-type: none"> Muestra una lista de comensales activos, con la posibilidad de buscar y el filtro de provincia, municipio, sexo, área, cargo, categoría, edificio, apartamento y residente. Seguido por una lista de personas con la posibilidad de Eliminar. 	
Observaciones: El usuario debe estar autenticado para realizar la acción.	
<p>Prototipo de interfaz:</p> 	

Historia de usuario	
Número: 29	Nombre: Gestionar permisos
Usuario: Administrador general	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración asignada: 1

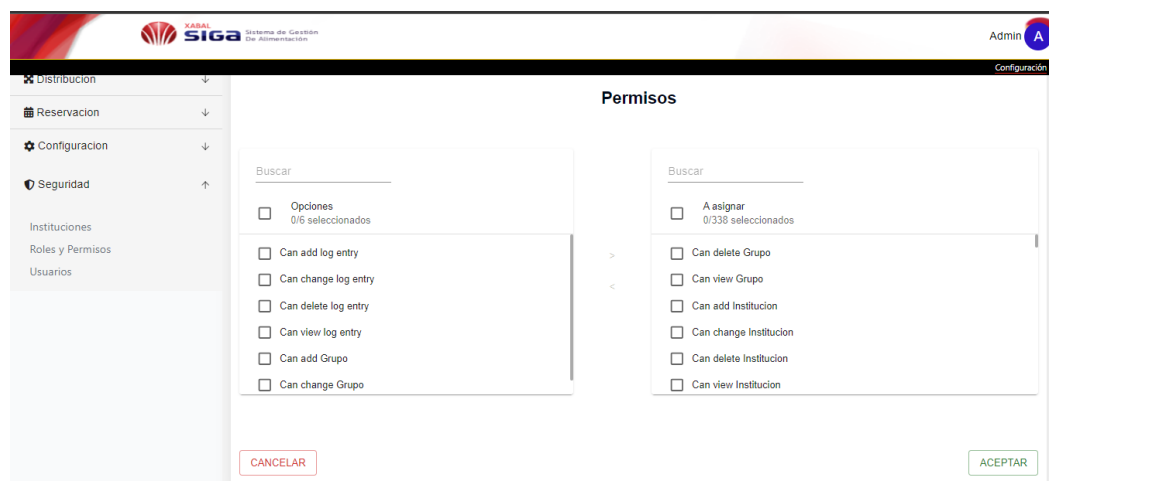
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

Descripción:

- Al editar un rol se muestra una lista en el lado izquierdo de la pantalla con los permisos que aún no han sido concedidos a este rol, y a mano derecha una lista con los permisos que ya posee. En el medio están dos botones uno para asignar, cuando se seleccione un permiso y otro para revocar cuando se seleccione un permiso de la lista de otorgados. Seguido por los botones aceptar y cancelar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



Historia de usuario

Número: 30

Nombre: Gestionar roles

Usuario: Administrador general

Prioridad en negocio: Alto

Riesgo en desarrollo: Alto

Puntos estimados: 0.5

Iteración asignada: 1

Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

Descripción:

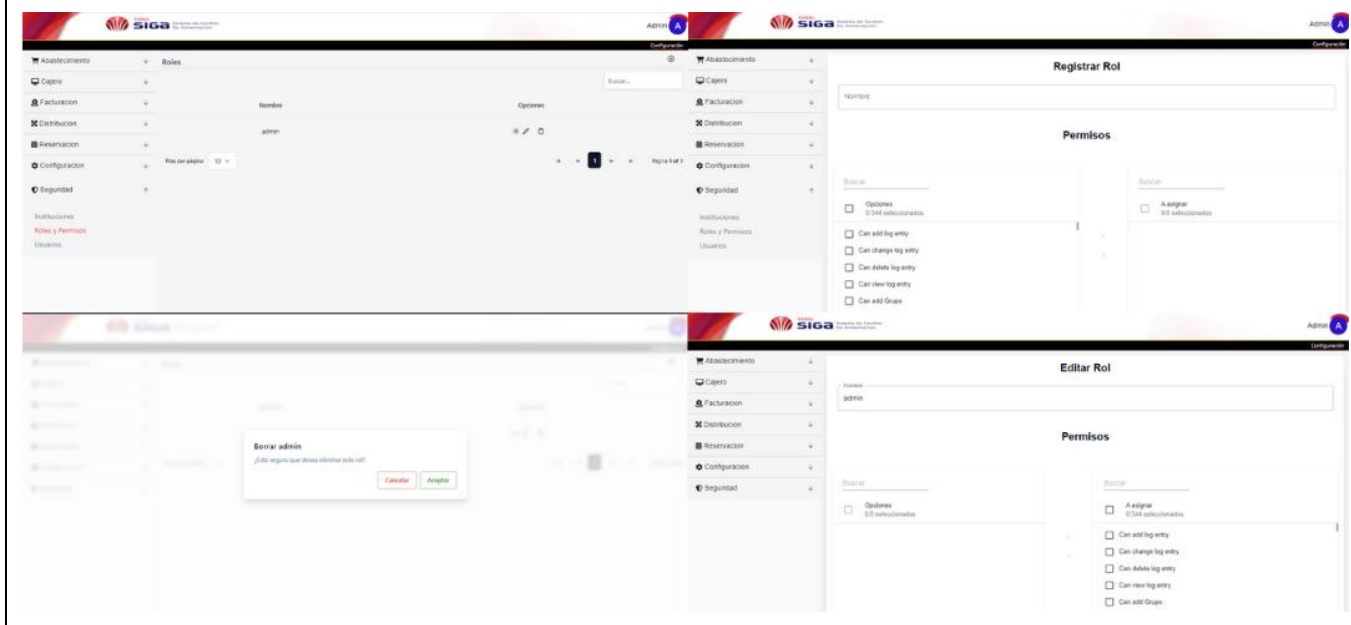
- Muestra una interfaz con la posibilidad de buscar y una lista de roles. Presenta la posibilidad de añadir, ver detalles, editar y eliminar.
- Registrar rol: se muestra una interfaz con un campo para el nombre y la posibilidad de gestionar los permisos, seguido por los botones aceptar y cancelar.
- Editar rol: se muestra una interfaz con un campo para el nombre y la posibilidad de gestionar

los permisos, seguido por los botones aceptar y cancelar.

- Eliminar se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



Historia de usuario

Número: 31

Nombre: Gestionar usuarios

Usuario: Administrador general

Prioridad en negocio: Alto

Riesgo en desarrollo: Alto

Puntos estimados: 0.6

Iteración asignada: 1

Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

Descripción:

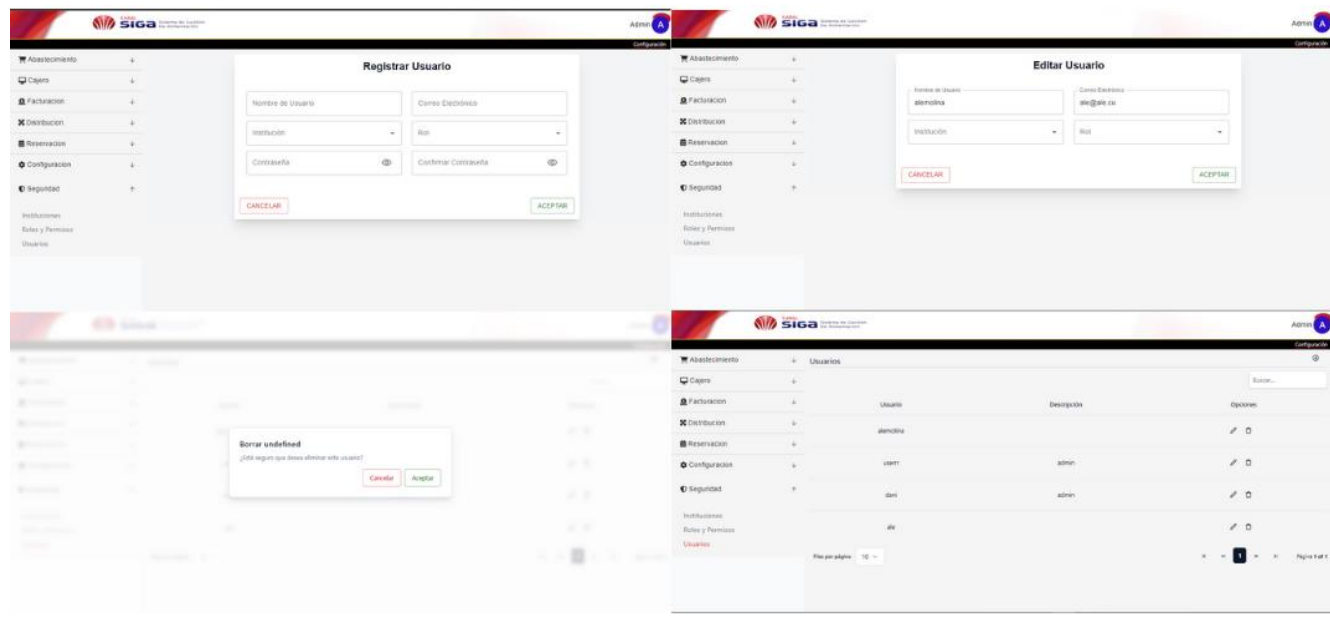
- Muestra una interfaz con la posibilidad de buscar. Seguido por una lista de usuarios con los campos usuario, descripción y la posibilidad de añadir, editar y eliminar.
- Registrar usuario se muestra una interfaz con los siguientes campos: rol, nombre de usuario, correo electrónico, institución, y contraseña y confirmar contraseña, y los botones aceptar y cancelar.
- Editar usuario se muestra una interfaz con los siguientes campos: rol, nombre de usuario,

correo electrónico, institución, y contraseña y confirmar contraseña, y los botones aceptar y cancelar.

- Eliminar se muestra una ventana emergente con el siguiente mensaje: ¿Está seguro que desea eliminar? Y los botones aceptar y cancelar.

Observaciones: El usuario debe estar autenticado para realizar la acción.

Prototipo de interfaz:



Historia de usuario

Número: 32

Nombre: Autenticación de usuarios

Usuario: Administrador general, Administrador institucional

Prioridad en negocio: Alto

Riesgo en desarrollo: Alto

Puntos estimados: 1

Iteración asignada: 1

Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila

Descripción: Muestra una pantalla con los campos usuario y contraseña y el botón “Entrar”.

Observaciones:

Prototipo de interfaz:



Anexo 2. Tarjetas CRC del módulo de configuración del sistema XABAL SIGA versión 2.0.

Nombre: TbNtipoProducto	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear un nuevo tipo de producto.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de tipos de producto.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar un tipo de producto.</p> <p>update():Este método maneja la solicitud PUT para actualizar un tipo de producto.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar parcialmente un tipo de producto.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar un tipo de producto.</p>	<p>TbInstitucion</p> <p>models.Model</p>

Nombre: TbDproducto	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear un nuevo producto.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de productos.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar un producto.</p> <p>update():Este método maneja la solicitud PUT para actualizar un producto.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar parcialmente un producto.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar un producto.</p>	<p>TbInstitucion</p> <p>models.Model</p> <p>tb_ntipo_producto</p> <p>TbNunidadMedida</p>

Nombre: TbNtipoTarjeta	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear un tipo de tarjeta.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de tipos de tarjeta.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar un tipo de tarjeta.</p> <p>update():Este método maneja la solicitud PUT para actualizar un tipo de tarjeta.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar parcialmente un tipo</p>	<p>TbInstitucion</p> <p>models.Model</p>

de tarjeta. destroy():Este método maneja la solicitud DELETE para eliminar un tipo de tarjeta.	
---	--

Nombre: TbDtarjetaAlimentacion	
Responsabilidad:	Colaboraciones:
create():Este método maneja la solicitud POST para crear una tarjeta. list():Este método maneja la solicitud GET para obtener una lista de tarjetas. retrieve():Este método maneja la solicitud GET para recuperar una tarjeta. update():Este método maneja la solicitud PUT para actualizar una tarjeta. partial_update():Este método maneja la solicitud PATCH para actualizar parcialmente una tarjeta. destroy():Este método maneja la solicitud DELETE para eliminar una tarjeta.	TbInstitucion models.Model TbNestadoTarjeta Usuario TbNtipoTarjeta

Nombre: TbRpersonaTarjeta	
Responsabilidad:	Colaboraciones:
create():Este método maneja la solicitud POST para crear un registro persona - tarjeta. list():Este método maneja la solicitud GET para obtener una lista de registro persona - tarjeta retrieve():Este método maneja la solicitud GET para recuperar	TbInstitucion models.Model TbNestadoTarjeta Usuario TbNtipoTarjeta

<p>un registro persona - tarjeta</p> <p>update():Este método maneja la solicitud PUT para actualizar un registro persona - tarjeta</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar parcialmente registro persona - tarjeta.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar un registro persona - tarjeta.</p>	
--	--

Nombre: TbDpersonalPPuerta	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear una asignación de IP a puerta.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de IP y de puertas.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar una puerta o un IP.</p> <p>update():Este método maneja la solicitud PUT para actualizar una puerta o un IP.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar una puerta y un IP.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar una asignación.</p>	<p>TbInstitucion</p> <p>models.Model</p> <p>TbDpersona</p> <p>TbStructure</p> <p>TbUser</p>

Nombre: TbDaccesoEventoSecundario	
Responsabilidad:	Colaboraciones:

<p>create():Este método maneja la solicitud POST para crear una cantidad de accesos.</p> <p>list():Este método maneja la solicitud GET para obtener una lista con la cantidad de accesos.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar una cantidad de accesos.</p> <p>.update():Este método maneja la solicitud PUT para actualizar una una cantidad de accesos.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar una cantidad de accesos.</p>	TbInstitucion
--	---------------

Nombre: TbDpersonaTorpedo	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear un torpedo.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de personas.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar un torpedo.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar un torpedo.</p>	<p>TbInstitucion</p> <p>models.Model</p> <p>TbUser</p>

Nombre: TbDsolapinPerdido	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear un solapín perdido.</p>	<p>TbInstitucion</p> <p>models.Model</p>

list():Este método maneja la solicitud GET para obtener una lista de personas.	TbUser
retrieve():Este método maneja la solicitud GET para recuperar un solapín perdido.	TbDpersona
destroy():Este método maneja la solicitud DELETE para eliminar un solapín perdido.	

Nombre: TbNestructura	
Responsabilidad:	Colaboraciones:
create():Este método maneja la solicitud POST para crear un área.	TbInstitucion
list():Este método maneja la solicitud GET para obtener una lista de áreas.	TbNtipoEstructura
update():Este método maneja la solicitud PUT para actualizar un área.	
partial_update():Este método maneja la solicitud PATCH para actualizar un área.	
destroy():Este método maneja la solicitud DELETE para eliminar un área.	

Nombre: TbNcategoria	
Responsabilidad:	Colaboraciones:
create():Este método maneja la solicitud POST para crear una categoría.	TbInstitucion
list():Este método maneja la solicitud GET para obtener una lista de categorías.	models.Model
retrieve():Este método maneja la solicitud GET para recuperar	

<p>una categoría.</p> <p>update():Este método maneja la solicitud PUT para actualizar una categoría.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar una categoría.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar una categoría.</p>	
--	--

Nombre: TbStructure	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear una configuración de estructura.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de configuraciones de estructuras.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar una configuración de estructuras.</p> <p>update():Este método maneja la solicitud PUT para actualizar una configuración de estructuras.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar una configuración de estructura.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar una configuración de estructuras.</p>	<p>TbInstitucion</p> <p>models.Model</p> <p>TbNtipoEstructura</p>

Nombre: TbNevento

Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear un evento.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de eventos.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar un evento.</p> <p>update():Este método maneja la solicitud PUT para actualizar un evento.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar un evento.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar un evento.</p>	<p>TbInstitucion</p> <p>models.Model</p> <p>TbNclasificacionEvento</p> <p>TbNhorario</p>

Nombre: TbNhorario	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear un horario.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de horarios.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar un horario.</p> <p>update():Este método maneja la solicitud PUT para actualizar un horario.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar un horario.</p> <p>destroy():Este método maneja la solicitud DELETE para</p>	<p>TbInstitucion</p> <p>models.Model</p>

eliminar un horario.	
----------------------	--

Nombre: TbNtipoEstructura	
Responsabilidad:	Colaboraciones:
create():Este método maneja la solicitud POST para crear un tipo de área.	TbInstitucion models.Model
list():Este método maneja la solicitud GET para obtener una lista de tipos de área.	TbNestructura
retrieve():Este método maneja la solicitud GET para recuperar un tipo de área.	
destroy():Este método maneja la solicitud DELETE para eliminar un tipo de área.	

Nombre: TbDperiodoReservacion	
Responsabilidad:	Colaboraciones:
list():Este método maneja la solicitud GET para obtener una lista de reservaciones.	TbInstitucion models.Model
partial_update():Este método maneja la solicitud PATCH para actualizar una reservación.	

Nombre: TbDelementosMostrar	
Responsabilidad:	Colaboraciones:
list():Este método maneja la solicitud GET para obtener una lista de elementos a mostrar.	TbInstitucion models.Model
partial_update():Este método maneja la solicitud PATCH para actualizar un elemento a mostrar.	

Nombre: TbDdatosContacto	
Responsabilidad:	Colaboraciones:
list():Este método maneja la solicitud GET para obtener una lista de datos de contacto.	TbInstitucion models.Model
partial_update():Este método maneja la solicitud PATCH para actualizar un dato de contacto.	Usuario

Nombre: TbDresponsableAreaPersonas	
Responsabilidad:	Colaboraciones:
create():Este método maneja la solicitud POST para crear una asociación de una persona a un área.	TbInstitucion models.Model Usuario
list():Este método maneja la solicitud GET para obtener una lista de personas.	TbDpersona TbNestructura
retrieve():Este método maneja la solicitud GET para recuperar una persona.	
update():Este método maneja la solicitud PUT para actualizar una asociación de una persona a un área.	
partial_update():Este método maneja la solicitud PATCH para actualizar una asociación de una persona a un área.	
destroy():Este método maneja la solicitud DELETE para eliminar una asociación de una persona a un área.	

Nombre: Institucion	
Responsabilidad:	Colaboraciones:
create():Este método maneja la solicitud POST para crear una	models.Model

<p>institución.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de instituciones.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar una institución.</p> <p>update():Este método maneja la solicitud PUT para actualizar una institución.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar una institución.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar una institución.</p>	
---	--

Nombre: TbDconfiguracionProceso	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear una configuración del proceso.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de configuraciones.</p> <p>update():Este método maneja la solicitud PUT para actualizar una configuración.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar una configuración.</p>	<p>models.Model</p> <p>Usuario</p> <p>TbInstitucion</p>

Nombre: TbNconfiguracionCobro	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear una regla de cobro.</p> <p>list():Este método maneja la</p>	<p>models.Model</p> <p>Usuario</p>

<p>solicitud GET para obtener una lista de configuraciones.</p> <p>update():Este método maneja la solicitud PUT para actualizar una configuración.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar una configuración.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar una configuración.</p>	<p>TbInstitucion</p> <p>TbTipoRegla</p>
--	---

Nombre: TbDconfiguracionPersona	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear una configuración de comensales.</p> <p>list():Este método maneja la solicitud GET para obtener una lista de configuraciones de comensales.</p> <p>update():Este método maneja la solicitud PUT para actualizar una configuración de comensales.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar una configuración de comensales.</p>	<p>Usuario</p> <p>TbInstitucion</p>

Nombre: Usuario	
Responsabilidad:	Colaboraciones:
<p>create():Este método maneja la solicitud POST para crear un usuario.</p> <p>list():Este método maneja la solicitud GET para obtener una</p>	<p>models.Model</p> <p>TbAvatar</p> <p>Usuario</p>

<p>lista de usuarios.</p> <p>retrieve():Este método maneja la solicitud GET para recuperar un usuario.</p> <p>update():Este método maneja la solicitud PUT para actualizar un usuario.</p> <p>partial_update():Este método maneja la solicitud PATCH para actualizar un usuario.</p> <p>destroy():Este método maneja la solicitud DELETE para eliminar un usuario.</p>	TbInstitucion
--	---------------

Nombre: CustomLoginView	
Responsabilidad:	Colaboraciones:
CustomLoginView	models.Model TbAvatar Usuario TbInstitucion

Anexo 3. Tareas de ingeniería por HU del módulo de configuración del sistema XABAL SIGA versión 2.0.

Tarea	
Número de la Tarea: 3	Número de la Historia de Usuario: 2
Nombre de la tarea: Desarrollar vista para el gestionar unidades de medida	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 28-06-2023	Fecha de fin: 29-06-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de unidades de medida.	

Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) las unidades de medida del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de unidades de medida.

Tarea	
Número de la Tarea: 4	Número de la Historia de Usuario: 2
Nombre de la tarea: Desarrollar funcionalidades para el gestionar unidades de medida	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 29-06-2023	Fecha de fin: 30-06-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) las unidades de medida. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar unidades de medida en la base de datos.	

Tarea	
Número de la Tarea: 5	Número de la Historia de Usuario: 3
Nombre de la tarea: Desarrollar vista para el gestionar tipos de productos	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3

Fecha de inicio: 30-06-2023	Fecha de fin: 3-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Se desarrollan las interfaces para la gestión de tipos de productos. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) los tipos de productos del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de tipos de productos.</p>	

Tarea	
Número de la Tarea: 6	Número de la Historia de Usuario: 3
Nombre de la tarea: Desarrollar funcionalidades para el gestionar tipos de productos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 3-07-2023	Fecha de fin: 4-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) los tipos de productos. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar tipos de productos en la base de datos.</p>	

Tarea	
Número de la Tarea: 7	Número de la Historia de Usuario: 4
Nombre de la tarea: Desarrollar vista para el gestionar productos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 5-07-2023	Fecha de fin: 6-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de productos. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) los productos del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de productos.	

Tarea	
Número de la Tarea: 8	Número de la Historia de Usuario: 4
Nombre de la tarea: Desarrollar funcionalidades para el gestionar productos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 6-07-2023	Fecha de fin: 10-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) los productos. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los	

usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar productos en la base de datos.

Tarea	
Número de la Tarea: 9	Número de la Historia de Usuario: 5
Nombre de la tarea: Desarrollar vista para el gestionar tipos de tarjetas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 10-07-2023	Fecha de fin: 11-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de tipos de tarjetas. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) los tipos de tarjetas del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de tipos de tarjetas.	

Tarea	
Número de la Tarea: 10	Número de la Historia de Usuario: 5
Nombre de la tarea: Desarrollar funcionalidades para el gestionar tipos de tarjetas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 11-07-2023	Fecha de fin: 12-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro	

Ávila.
<p>Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) los tipos de tarjetas. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar tipos de tarjetas en la base de datos.</p>

Tarea	
Número de la Tarea: 11	Número de la Historia de Usuario: 6
Nombre de la tarea: Desarrollar vista para el gestionar tarjeta.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 12-07-2023	Fecha de fin: 13-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Se desarrollan las interfaces para la gestión de tarjetas. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) las tarjetas del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de tarjetas.</p>	

Tarea	
Número de la Tarea: 12	Número de la Historia de Usuario: 6
Nombre de la tarea: Desarrollar funcionalidades para el gestionar tarjetas.	

Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 13-07-2023	Fecha de fin: 14-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) tarjetas. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar tarjetas en la base de datos.	

Tarea	
Número de la Tarea: 13	Número de la Historia de Usuario: 7
Nombre de la tarea: Desarrollar vista para el asociar tarjeta a usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 14-07-2023	Fecha de fin: 17-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la asociación de tarjetas a usuarios. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer y actualizar las asociaciones de tarjetas al usuario del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización y modificación de asociaciones.	

Tarea	
Número de la Tarea: 14	Número de la Historia de Usuario: 7
Nombre de la tarea: Desarrollar funcionalidades para el asociar tarjetas a usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 14-07-2023	Fecha de fin: 17-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer y actualizar asociaciones de tarjetas a usuarios. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar y actualizar asociaciones de tarjetas a usuarios en la base de datos.	

Tarea	
Número de la Tarea: 15	Número de la Historia de Usuario: 8
Nombre de la tarea: Desarrollar vista para el gestionar asignación de IP a puerta.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 17-07-2023	Fecha de fin: 18-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de asignación de IP a puertas. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) las asignaciones de IP a puerta	

del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de asignaciones.

Tarea	
Número de la Tarea: 16	Número de la Historia de Usuario: 8
Nombre de la tarea: Desarrollar funcionalidades para el gestionar asignación de IP a puerta.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 18-07-2023	Fecha de fin: 20-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) asignaciones de IP a puertas. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar asignaciones de IP a puertas en la base de datos.	
Tarea	
Número de la Tarea: 17	Número de la Historia de Usuario: 9
Nombre de la tarea: Desarrollar vista para configurar cantidad de accesos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 20-07-2023	Fecha de fin: 21-07-2023

Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.
Descripción: Se desarrollan las interfaces para la configuración de cantidad de accesos. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para leer y actualizar las configuraciones de cantidad de accesos del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la visualización y modificación de la configuración de cantidad de accesos.

Tarea	
Número de la Tarea: 18	Número de la Historia de Usuario: 9
Nombre de la tarea: Desarrollar funcionalidades para el configurar cantidad de accesos	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 21-07-2023	Fecha de fin: 24-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias leer y actualizar configuraciones de cantidad de acceso. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para visualizar y actualizar configuraciones de cantidad de accesos en la base de datos.	

Tarea

Número de la Tarea: 19	Número de la Historia de Usuario: 10
Nombre de la tarea: Desarrollar vista para el gestionar torpedos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 24-07-2023	Fecha de fin: 25-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Se desarrollan las interfaces para la gestión de torpedos. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) los torpedos del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de torpedos.</p>	

Tarea	
Número de la Tarea: 20	Número de la Historia de Usuario: 10
Nombre de la tarea: Desarrollar funcionalidades para el gestionar torpedos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 25-07-2023	Fecha de fin: 26-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) torpedos. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y contro-</p>	

ladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar torpedos en la base de datos.

Tarea	
Número de la Tarea: 21	Número de la Historia de Usuario: 11
Nombre de la tarea: Desarrollar vista para el administrar solapín perdido.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 26-07-2023	Fecha de fin: 27-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Se desarrollan las interfaces para la administración de solapines perdidos. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer y eliminar los solapines perdidos del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización y eliminación de solapines perdidos.</p>	

Tarea	
Número de la Tarea: 22	Número de la Historia de Usuario: 11
Nombre de la tarea: Desarrollar funcionalidades para el administrar solapines perdidos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 27-07-2023	Fecha de fin: 28-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro	

Ávila.
<p>Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer y eliminar solapines perdidos. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, buscar y eliminar solapines perdidos en la base de datos.</p>

Tarea	
Número de la Tarea: 23	Número de la Historia de Usuario: 12
Nombre de la tarea: Desarrollar vista para el gestionar áreas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 28-07-2023	Fecha de fin: 31-07-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Se desarrollan las interfaces para la gestión de áreas. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) las áreas del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de áreas.</p>	

Tarea	
Número de la Tarea: 24	Número de la Historia de Usuario: 12
Nombre de la tarea: Desarrollar funcionalidades para el gestionar áreas.	

Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 31-07-2023	Fecha de fin: 1-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) áreas. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar áreas en la base de datos.	

Tarea	
Número de la Tarea: 25	Número de la Historia de Usuario: 13
Nombre de la tarea: Desarrollar vista para el gestionar categorías.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 1-08-2023	Fecha de fin: 2-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de categorías. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) las categorías del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de categorías.	

Tarea	
Número de la Tarea: 26	Número de la Historia de Usuario: 13
Nombre de la tarea: Desarrollar funcionalidades para el gestionar categorías	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 2-08-2023	Fecha de fin: 3-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) categorías. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar categorías en la base de datos.</p>	

Tarea	
Número de la Tarea: 27	Número de la Historia de Usuario: 14
Nombre de la tarea: Desarrollar vista para el gestionar estructuras.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 3-08-2023	Fecha de fin: 4-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Se desarrollan las interfaces para la gestión de estructuras. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) las estructuras del módulo de configuración. El</p>	

objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de estructuras.

Tarea	
Número de la Tarea: 28	Número de la Historia de Usuario: 14
Nombre de la tarea: Desarrollar funcionalidades para el gestionar estructuras	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 4-08-2023	Fecha de fin: 7-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) estructuras. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar estructuras en la base de datos.	

Tarea	
Número de la Tarea: 29	Número de la Historia de Usuario: 15
Nombre de la tarea: Desarrollar vista para el configurar estructuras.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 7-08-2023	Fecha de fin: 8-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro	

Ávila.
<p>Descripción: Se desarrollan la interfaz para la configuración de estructuras. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para leer y actualizar las configuraciones de estructuras del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la visualización y modificación de configuraciones de estructuras.</p>

Tarea	
Número de la Tarea: 30	Número de la Historia de Usuario: 15
Nombre de la tarea: Desarrollar funcionalidades para el configurar estructuras	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 8-08-2023	Fecha de fin: 9-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para leer y actualizar configuraciones de estructuras. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para recuperar y actualizar configuraciones de estructuras en la base de datos.</p>	

Tarea	
Número de la Tarea: 31	Número de la Historia de Usuario: 16
Nombre de la tarea: Desarrollar vista para el gestionar eventos.	

Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 9-08-2023	Fecha de fin: 10-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de eventos. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) los eventos del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de eventos.	

Tarea	
Número de la Tarea: 32	Número de la Historia de Usuario: 16
Nombre de la tarea: Desarrollar funcionalidades para el gestionar eventos	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 10-08-2023	Fecha de fin: 11-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) eventos. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar eventos en la base de datos.	

Tarea	
Número de la Tarea: 33	Número de la Historia de Usuario: 17
Nombre de la tarea: Desarrollar vista para el gestionar horarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 11-08-2023	Fecha de fin: 14-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de horarios. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) los horarios del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de horarios.	

Tarea	
Número de la Tarea: 34	Número de la Historia de Usuario: 17
Nombre de la tarea: Desarrollar funcionalidades para el gestionar horarios	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 14-08-2023	Fecha de fin: 15-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) horarios. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios reali-	

zar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar horarios en la base de datos.

Tarea	
Número de la Tarea: 35	Número de la Historia de Usuario: 18
Nombre de la tarea: Desarrollar vista para el gestionar tipos de áreas	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 15-08-2023	Fecha de fin: 16-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de tipos de áreas. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer y eliminar los tipos de áreas. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización y eliminación de tipos de áreas.	

Tarea	
Número de la Tarea: 36	Número de la Historia de Usuario: 18
Nombre de la tarea: Desarrollar funcionalidades para el administrar configuración de reglas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 15-08-2023	Fecha de fin: 16-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro	

Ávila.
<p>Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer y eliminar tipos de áreas. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, buscar y eliminar tipos de áreas en la base de datos.</p>

Tarea	
Número de la Tarea: 37	Número de la Historia de Usuario: 19
Nombre de la tarea: Desarrollar vista para el administrar períodos de reservación	
Tipo de tarea: Desarrollo	Puntos estimados: 0,1
Fecha de inicio: 16-08-2023	Fecha de fin: 17-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Se desarrolla la interfaz para la administración de períodos de reservación. Esta tarea implica el diseño y desarrollo de la interfaz de usuarios necesaria para leer y modificar los períodos de reservación del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la visualización y modificación de períodos.</p>	

Tarea	
Número de la Tarea: 38	Número de la Historia de Usuario: 19
Nombre de la tarea: Desarrollar funcionalidades para el administrar períodos de	

reservación.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 16-08-2023	Fecha de fin: 17-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para leer y modificar períodos de reservación. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para recuperar y modificar períodos de reservación en la base de datos.</p>	

Tarea	
Número de la Tarea: 39	Número de la Historia de Usuario: 20
Nombre de la tarea: Desarrollar vista para el administrar elementos a mostrar	
Tipo de tarea: Desarrollo	Puntos estimados: 0,1
Fecha de inicio: 17-08-2023	Fecha de fin: 18-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Se desarrolla la interfaz para la administración de elementos a mostrar. Esta tarea implica el diseño y desarrollo de la interfaz de usuarios necesaria para leer y modificar los elementos a mostrar del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la visualización y modificación de elementos a mostrar.</p>	

Tarea	
Número de la Tarea: 40	Número de la Historia de Usuario: 20
Nombre de la tarea: Desarrollar funcionalidades para el administrar elementos a mostrar	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 17-08-2023	Fecha de fin: 18-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para leer y modificar elementos a mostrar. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para recuperar y modificar elementos a mostrar en la base de datos.	

Tarea	
Número de la Tarea: 41	Número de la Historia de Usuario: 21
Nombre de la tarea: Desarrollar vista para el administrar datos de contacto	
Tipo de tarea: Desarrollo	Puntos estimados: 0,1
Fecha de inicio: 18-08-2023	Fecha de fin: 21-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	

Descripción: Se desarrolla la interfaz para la administración de datos de contacto. Esta tarea implica el diseño y desarrollo de la interfaz de usuarios necesaria para leer y modificar los datos de contacto del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la visualización y modificación de datos de contacto.

Tarea	
Número de la Tarea: 42	Número de la Historia de Usuario: 21
Nombre de la tarea: Desarrollar funcionalidades para el administrar datos de contacto	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 18-08-2023	Fecha de fin: 21-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para leer y modificar datos de contacto. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para recuperar y modificar datos de contacto en la base de datos.	

Tarea	
Número de la Tarea: 43	Número de la Historia de Usuario: 22
Nombre de la tarea: Desarrollar vista para el administrar responsables de áreas	

Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 21-08-2023	Fecha de fin: 22-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Se desarrolla la interfaz para la administración de responsables de áreas. Esta tarea implica el diseño y desarrollo de la interfaz de usuarios necesaria para leer y modificar los responsables de áreas del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la visualización y modificación de responsables de áreas.</p>	

Tarea	
Número de la Tarea: 44	Número de la Historia de Usuario: 22
Nombre de la tarea: Desarrollar funcionalidades para el administrar responsables de áreas	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 21-08-2023	Fecha de fin: 22-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para leer y modificar responsables de áreas. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para recuperar y modificar responsables de áreas en la base de datos.</p>	

Tarea	
Número de la Tarea: 45	Número de la Historia de Usuario: 23
Nombre de la tarea: Desarrollar vista para el gestionar módulos por institución.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 22-08-2023	Fecha de fin: 23-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de módulos. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) los módulos por institución del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de módulos.	

Tarea	
Número de la Tarea: 46	Número de la Historia de Usuario: 23
Nombre de la tarea: Desarrollar funcionalidades para el gestionar módulos por instituciones	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 23-08-2023	Fecha de fin: 24-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) módulos. El objetivo principal	

es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar módulos por instituciones.

Tarea	
Número de la Tarea: 47	Número de la Historia de Usuario: 24
Nombre de la tarea: Desarrollar vista para el gestionar instituciones	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 24-08-2023	Fecha de fin: 25-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de instituciones. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) las instituciones del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de instituciones.	

Tarea	
Número de la Tarea: 48	Número de la Historia de Usuario: 24
Nombre de la tarea: Desarrollar funcionalidades para el gestionar instituciones	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 25-08-2023	Fecha de fin: 28-08-2023

Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) instituciones. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar instituciones.

Tarea	
Número de la Tarea: 49	Número de la Historia de Usuario: 25
Nombre de la tarea: Desarrollar vista para el administrar configuraciones del proceso	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 28-08-2023	Fecha de fin: 29-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrolla la interfaz para la administración de configuraciones del proceso. Esta tarea implica el diseño y desarrollo de la interfaz de usuarios necesaria para leer y modificar las configuraciones del proceso. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la visualización y modificación de configuraciones del proceso.	

Tarea

Número de la Tarea: 50	Número de la Historia de Usuario: 25
Nombre de la tarea: Desarrollar funcionalidades para el administrar configuraciones del proceso	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 29-08-2023	Fecha de fin: 30-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para leer y modificar configuraciones del proceso. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para recuperar y modificar configuraciones del proceso en la base de datos.	

Tarea	
Número de la Tarea: 51	Número de la Historia de Usuario: 26
Nombre de la tarea: Desarrollar vista para el gestionar configuración del cobro	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 30-08-2023	Fecha de fin: 31-08-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de configuración del cobro. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) las configuraciones de cobro del mó-	

dulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de configuraciones de cobro.

Tarea	
Número de la Tarea: 52	Número de la Historia de Usuario: 26
Nombre de la tarea: Desarrollar funcionalidades para el gestionar configuración de cobro	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 31-08-2023	Fecha de fin: 1-09-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) configuraciones de cobro. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar configuraciones de cobro.	

Tarea	
Número de la Tarea: 53	Número de la Historia de Usuario: 27
Nombre de la tarea: Desarrollar vista para el administrar configuraciones de comensales	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3

Fecha de inicio: 1-09-2023	Fecha de fin: 4-09-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrolla la interfaz para la administración de configuraciones de comensales. Esta tarea implica el diseño y desarrollo de la interfaz de usuarios necesaria para leer y adicionar las configuraciones de comensales. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la visualización y modificación de configuraciones de comensales.	

Tarea	
Número de la Tarea: 54	Número de la Historia de Usuario: 27
Nombre de la tarea: Desarrollar funcionalidades para el administrar configuraciones de comensales	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 4-09-2023	Fecha de fin:5-09-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para leer y modificar configuraciones de comensales. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para recuperar y modificar configuraciones de comensales en la base de datos.	

Tarea	
Número de la Tarea: 55	Número de la Historia de Usuario: 28
Nombre de la tarea: Desarrollar vista para el administrar comensales	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 5-09-2023	Fecha de fin: 6-09-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrolla la interfaz para la administración de comensales. Esta tarea implica el diseño y desarrollo de la interfaz de usuarios necesaria para leer y eliminar comensales. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la visualización y eliminación de comensales.	

Tarea	
Número de la Tarea: 56	Número de la Historia de Usuario: 28
Nombre de la tarea: Desarrollar funcionalidades para el administrar comensales	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 6-09-2023	Fecha de fin:7-09-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para leer y eliminar comensales. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen	

las solicitudes de los usuarios para recuperar y eliminar comensales en la base de datos.

Tarea	
Número de la Tarea: 57	Número de la Historia de Usuario: 29
Nombre de la tarea: Desarrollar vista para el gestionar permisos	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 7-09-2023	Fecha de fin: 8-09-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
Descripción: Se desarrollan las interfaces para la gestión de permisos. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) los permisos del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de permisos.	

Tarea	
Número de la Tarea: 58	Número de la Historia de Usuario: 29
Nombre de la tarea: Desarrollar funcionalidades para el gestionar permisos	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 8-09-2023	Fecha de fin: 11-09-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro	

Ávila.
<p>Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) permisos. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar permisos.</p>

Tarea	
Número de la Tarea: 59	Número de la Historia de Usuario: 30
Nombre de la tarea: Desarrollar vista para el gestionar roles	
Tipo de tarea: Desarrollo	Puntos estimados: 0,2
Fecha de inicio: 11-09-2023	Fecha de fin: 12-09-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Se desarrollan las interfaces para la gestión de roles. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) los roles del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de roles.</p>	

Tarea	
Número de la Tarea: 60	Número de la Historia de Usuario: 30

Nombre de la tarea: Desarrollar funcionalidades para el gestionar roles	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 12-09-2023	Fecha de fin: 13-09-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) roles. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar roles.</p>	

Tarea	
Número de la Tarea: 61	Número de la Historia de Usuario: 31
Nombre de la tarea: Desarrollar vista para el gestionar permisos	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 13-09-2023	Fecha de fin: 15-09-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Se desarrollan las interfaces para la gestión de usuarios. Esta tarea implica el diseño y desarrollo de las interfaces de usuario necesarias para crear, leer, actualizar y eliminar (CRUD) los usuarios del módulo de configuración. El objetivo es proporcionar a los usuarios una experiencia intuitiva y eficiente. Esto incluye la creación de formularios que permitan la adición, búsqueda, visualización, modificación y eliminación de usuarios.</p>	

Tarea	
Número de la Tarea: 62	Número de la Historia de Usuario: 31
Nombre de la tarea: Desarrollar funcionalidades para el gestionar usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Fecha de inicio: 15-09-2023	Fecha de fin: 21-09-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Esta tarea implica la implementación de las funcionalidades clave necesarias para crear, leer, actualizar y eliminar (CRUD) usuarios. El objetivo principal es desarrollar el backend y las lógicas de negocio que permitan a los usuarios realizar operaciones eficientes. Esto incluye la creación de serializadores y controladores que gestionen las solicitudes de los usuarios para agregar, recuperar, actualizar, buscar y eliminar usuarios.</p>	

Tarea	
Número de la Tarea: 63	Número de la Historia de Usuario: 32
Nombre de la tarea: Desarrollar vista para el autenticar usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 0,5
Fecha de inicio: 21-09-2023	Fecha de fin: 28-09-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Esta tarea se centra en el diseño y desarrollo de la interfaz de usuario destinada a facilitar el proceso de autenticación de usuarios en un sistema o aplica-</p>	

ción. El objetivo principal es crear una interfaz que permita a los usuarios iniciar sesión de manera segura y eficiente. Esto implica la creación de pantallas de inicio de sesión que incluyan campos para ingresar credenciales como nombre de usuario y contraseña, así como la implementación de mecanismos de autenticación, como encriptado de contraseñas y sesiones seguras. El resultado final será una interfaz que garantice la autenticación de usuarios de manera efectiva, sin comprometer la seguridad del sistema.

Tarea	
Número de la Tarea: 64	Número de la Historia de Usuario: 32
Nombre de la tarea: Desarrollar funcionalidades para el autenticar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0,5
Fecha de inicio: 28-09-2023	Fecha de fin: 6-10-2023
Programadores responsables: Alejandro Javier Molina Santos, Enrique Ferreiro Ávila.	
<p>Descripción: Esta tarea se enfoca en desarrollar las funcionalidades esenciales para la autenticación de usuarios en un sistema o aplicación. El objetivo principal consiste en crear un sistema de autenticación sólido y seguro que habilite a los usuarios a registrar cuentas, iniciar sesiones y administrar sus perfiles de manera eficiente. Esto implica la implementación de servicios, controladores y lógica de backend encargados de gestionar la creación y verificación de credenciales de usuario, como nombres de usuario y contraseñas. El resultado final se traducirá en una garantía de seguridad y privacidad para los usuarios, permitiéndoles acceder de forma segura a las funcionalidades y datos del sistema sin comprometer la integridad de sus cuentas.</p>	

Anexo 4 Pruebas de aceptación al módulo de configuración del sistema XABAL SIGA Versión 2.0.

Caso de prueba de aceptación	
Código:H3_PA3	Historia de Usuario:3
Nombre: Gestionar tipos de productos	
Descripción: Prueba de las funcionalidades adicionar, eliminar, modificar y buscar tipos de productos.	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe estar autenticado • El usuario debe encontrarse en el módulo de configuración 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Presiona abastecimiento. 2. Presiona tipos de productos 3. Se carga la vista con el listado de tipos de productos <p>Adicionar:</p> <ol style="list-style-type: none"> 1. Presiona en Adicionar tipos de productos 2. Se muestra una interfaz con los campos: nombre, activo y descripción. Seguido por los botones Aceptar y Cancelar. 3. Se llenan los campos 4. Presiona botón Aceptar 5. Se muestra una notificación con el mensaje: "Tipo de producto creado". <p>Eliminar</p> <ol style="list-style-type: none"> 1. Presiona el botón Eliminar 2. Se muestra una interfaz con el mensaje: "¿Está seguro que desea eliminar este tipo de producto?". 	

3. Presiona el botón Aceptar.
4. Se muestra una notificación con el mensaje: "Tipo de producto eliminado"

Modificar

1. Presiona en Editar tipos de productos
2. Se muestra una interfaz con los campos: nombre, activo y descripción. Seguido por los botones Aceptar y Cancelar.

3. Se llenan los campos
4. Presiona botón Aceptar.
5. Se muestra una notificación con el mensaje: "Tipo de producto eliminado".

Buscar

1. Llena el campo de búsqueda
2. Se filtran los datos de la lista.

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H4_PA4

Historia de Usuario: 4

Nombre: Gestionar productos

Descripción: Prueba de las funcionalidades adicionar, eliminar, modificar y buscar productos.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona abastecimiento.
2. Presiona productos
3. Se carga la vista con el listado de productos

Adicionar:

1. Presiona en Adicionar productos
2. Se muestra una interfaz con los campos: nombre, descripción, precio, tipo de producto y unidad de medida. Seguido por los botones aceptar y cancelar.
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Producto creado"

Eliminar

1. Presiona el botón Eliminar
2. Se muestra una interfaz con el mensaje: "¿Está seguro que desea eliminar este producto?".
3. Presiona el botón Aceptar
4. Se muestra una notificación con el mensaje: "Producto eliminado".

Modificar

1. Presiona en Editar productos
2. Se muestra una interfaz con los campos: nombre, descripción, precio, tipo de producto y unidad de medida. Seguido por los botones aceptar y cancelar.
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Producto editado".

Buscar

1. Llena el campo de búsqueda
2. Se filtran los datos de la lista.

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H5_PA5

Historia de Usuario: 5

Nombre: Gestionar tipos de tarjetas

Descripción: Prueba de las funcionalidades adicionar, eliminar, modificar y buscar tipos de tarjetas.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona cajero.
2. Presiona tipos de tarjetas
3. Se carga la vista con el listado de tipos de tarjetas

Adicionar:

1. Presiona en Adicionar tipos de tarjetas
2. Se muestra una interfaz con los campos: nombre, descripción, color, activo. Seguido por los botones aceptar y cancelar.
3. Se llenan los campos
4. Presiona botón Aceptar

5. Se muestra una notificación con el mensaje: "Tipo de tarjeta creada".

Eliminar

1. Presiona el botón Eliminar
2. Se muestra una interfaz con el mensaje: "¿Está seguro que desea eliminar este tipo de tarjeta?".
3. Presiona el botón Aceptar
4. Se muestra una notificación con el mensaje: "Tipo de tarjeta eliminada".

Modificar

1. Presiona en Editar tipos de tarjetas.
2. Se muestra una interfaz con los campos: nombre, descripción, color, activo. Seguido por los botones aceptar y cancelar
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Tipo de tarjeta editada".

Buscar

1. Llena el campo de búsqueda
2. Se filtran los datos de la lista.

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H6_PA6

Historia de Usuario: 6

Nombre: Gestionar tarjetas

Descripción: Prueba de las funcionalidades adicionar, editar estado, eliminar, modificar y buscar tarjetas.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona cajero.
2. Presiona tarjetas
3. Se carga la vista con el listado de tarjetas

Adicionar:

1. Presiona en Adicionar tarjetas
2. Se muestra una interfaz con los campos fecha inicio, fin, estado y tipo de tarjeta. Seguido por los botones aceptar y cancelar.
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Tarjeta creada".

Eliminar

1. Presiona el botón Eliminar
2. Muestra una interfaz con el mensaje: "¿Está seguro que desea eliminar esta tarjeta?"
3. Presiona el botón Aceptar
4. Se muestra una notificación con el mensaje: "Tarjeta eliminada".

Modificar

1. Presiona en Editar tarjetas
2. Se muestra una interfaz con los campos: fecha inicio, fin, estado y tipo de tarjeta. Seguido por los botones aceptar y cancelar.

3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Tarjeta editada".

Buscar

1. Llena el campo de búsqueda
2. Se filtran las tarjetas de la lista.

Editar estado

1. Presiona editar estado
2. Se muestra una interfaz con el campo de estado seguido por los botones aceptar y cancelar.
3. Llena los campos
4. Presiona Aceptar
5. Se muestra una notificación con el mensaje: "Tarjeta editada".

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H7_PA7

Historia de Usuario: 7

Nombre: Asociar tarjeta a usuario

Descripción: Prueba de la funcionalidad de asociar tarjeta a persona.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona cajero.
2. Presiona tarjetas
3. Presiona Asociar tarjeta
4. Se carga la vista con el listado de personas
5. Presiona el botón asignar responsable
6. Se muestra una notificación con el mensaje: "Persona asociada correctamente".

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H8_PA8

Historia de Usuario: 8

Nombre: Gestionar asignación de IP a puerta

Descripción: Prueba de las funcionalidades adicionar, eliminar y modificar asignaciones de IP a puerta.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona cajero.
2. Presiona asignar IP a puerta
3. Se carga la vista con el listado de puertas

Asignar IP:

1. Presiona en Asignar IP a puerta
2. Se muestran los detalles de la tarjeta seguido por un campo para registrar el IP y el botón Asociar.
3. Se llenan los campos
4. Presiona botón Asociar
5. Se muestra una notificación con el mensaje: "IP asignado".

Eliminar

1. Presiona asignar IP
2. Presiona el botón Eliminar
3. Se muestra una interfaz con el mensaje "¿Está seguro que desea eliminar este IP?".
4. Presiona el botón Aceptar
5. Se muestra una notificación con el mensaje: "IP eliminado".

Modificar

1. Presiona asignar IP
2. Presiona en Editar IP
3. Se modifica el IP en el campo IP.
4. Presiona botón Editar
5. Se muestra una notificación con el mensaje "IP editado".

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H9_PA9

Historia de Usuario: 9

Nombre: Prueba de funcionalidad de configurar cantidad de accesos

Descripción: Prueba de la funcionalidad de configurar la cantidad de accesos posibles al servicio de alimentación.
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario debe estar autenticado • El usuario debe encontrarse en el módulo de configuración
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Presiona cajero. 2. Presiona configuración de acceso. 3. Se carga la vista con las configuraciones de accesos y el botón Establecer por defecto. 4. Presiona establecer por defecto. 5. Se carga una vista con una fecha y cantidad de accesos y el botón editar. 6. Presiona en Editar. 7. Se carga una vista que permite Editar Accesos, un campo para definir un valor y el botón Editar. 8. Llena los campos 9. Presiona editar. 10. Muestra una notificación con el mensaje: "Cantidad de acceso editada".
Resultado: Satisfactorio.

Caso de prueba de aceptación	
Código:H10_PA10	Historia de Usuario: 10
Nombre: Gestionar torpedos	

Descripción: Prueba de las funcionalidades adicionar, eliminar y buscar torpedos.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona cajero.
2. Presiona torpedos.
3. Se carga la vista con el listado de torpedos.

Adicionar:

1. Presiona en Adicionar Torpedos.
2. Se muestra una interfaz con los campos: nombre, CI, descripción, sexo, municipio, provincia y país. Seguido por los botones Aceptar y Cancelar.
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Torpedo creado".

Eliminar

1. Presiona el botón Eliminar.
2. Se muestra una interfaz con el siguiente mensaje: "¿Está seguro que desea eliminar este torpedo?".
3. Presiona el botón Aceptar
4. Se muestra una notificación con el mensaje: "Torpedo eliminado".

Buscar

1. Llena el campo de búsqueda

2. Se filtra la lista según lo buscado.

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H11_PA11

Historia de Usuario: 11

Nombre: Administrar solapín perdido

Descripción: Prueba de las funcionalidades adicionar, eliminar y buscar solapines perdidos.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona cajero.
2. Presiona solapín perdido.
3. Se carga la vista con el listado de personas con solapines perdidos.
Adicionar:
 1. Presiona en Adicionar
 2. Se muestra un listado de personas con la posibilidad de seleccionarlas
 3. Presiona el botón registrar solapin perdido.
 4. Se muestra una interfaz con el siguiente mensaje: “¿Está seguro que desea registrar este Solapín Perdido?”
 5. Presiona botón Aceptar
 6. Se muestra una notificación con el siguiente mensaje: “Persona con solapín

perdido añadida”.

Eliminar

1. Presiona el botón Eliminar
2. Se muestra una interfaz con el mensaje: “¿Está seguro que desea eliminar este Solapín Perdido?”.
3. Presiona el botón Aceptar
4. Se muestra una notificación con el mensaje: “Persona con solapín perdido eliminada”.

Buscar

1. Llena el campo de búsqueda
2. Se filtra la lista según lo buscado.

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H12_PA12

Historia de Usuario: 12

Nombre: Gestionar áreas.

Descripción: Prueba de las funcionalidades adicionar, eliminar, editar y buscar áreas.

Condiciones de ejecución:

- El usuario debe estar autenticado.
- El usuario debe encontrarse en el módulo de configuración.

Pasos de ejecución:

1. Presiona configuración.
2. Presiona configuración de área.

3. Se carga la vista con el listado de áreas.

Adicionar:

1. Presiona en Adicionar
2. Se muestra una interfaz con los campos: nombre, código área, código externo, área credencial, área consejo, activo y tipo de área. Seguido por los botones Aceptar y Cancelar.
3. Llena los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Área creada".

Editar

1. Presiona editar
2. Se muestra una interfaz con los campos: nombre, código área, código externo, área credencial, área consejo, activo y tipo de área. Seguido por los botones Aceptar y Cancelar.
3. Llena los campos.
4. Presiona botón Aceptar.
5. Se muestra una notificación con el mensaje: "Área editada".

Eliminar

1. Presiona el botón Eliminar
2. Se muestra una interfaz con el mensaje: ¿Está seguro que desea eliminar esta área?
3. Presiona el botón Aceptar
4. Se muestra una notificación con el mensaje: "Área eliminada".

Buscar

1. Llena el campo de búsqueda.

2. Se filtra la lista según lo buscado.

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H13_PA13

Historia de Usuario: 13

Nombre: Gestionar categorías

Descripción: Prueba de las funcionalidades adicionar, eliminar, modificar y buscar categorías.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona distribución.
2. Presiona categorías.
3. Se carga la vista con el listado de categorías

Adicionar:

1. Presiona en Adicionar
2. Se muestra una interfaz con los siguientes campos: nombre, descripción, color, activo y categoría. Seguido por los botones Aceptar y Cancelar.
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Categoría creada".

Eliminar

1. Presiona el botón Eliminar
2. Se muestra una interfaz con el siguiente mensaje: ¿Está seguro que desea eliminar esta categoría?.
3. Presiona el botón Aceptar
4. Se muestra una notificación con el siguiente mensaje: "Categoría eliminada".

Modificar

1. Presiona en Editar
2. Se muestra una interfaz con los siguientes campos: nombre, descripción, color, activo y categoría. Seguido por los botones Aceptar y Cancelar.
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Categoría editada".

Buscar

1. Llena el campo de búsqueda
2. Se filtra la lista según lo buscado.

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H14_PA14

Historia de Usuario: 14

Nombre: Gestionar estructuras

Descripción: Prueba de las funcionalidades adicionar, eliminar, editar y ver detalles.

Condiciones de ejecución:

- El usuario debe estar autenticado

- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona distribución.
2. Presiona estructuras.
3. Se carga la vista con una lista de estructuras de forma jerárquica.

Adicionar:

1. Presiona en Adicionar
2. Se muestra una interfaz con los campos: nombre, iniciales, capacidad, descripción, activo y categoría. Seguido por los botones Aceptar y cancelar.
3. Se llenan los campos
4. Presiona botón Siguiente.
5. Se muestra una interfaz para seleccionar la estructura padre.
6. Se selecciona
7. Presiona el botón Aceptar
8. Se muestra una notificación con el mensaje: "Estructura creada".

Eliminar

1. Presiona el botón Eliminar
2. Se muestra una interfaz con el mensaje: "¿Está seguro que desea eliminar esta estructura?"
3. Presiona el botón Aceptar
4. Se muestra una notificación con el mensaje: "Estructura eliminada"

Modificar

1. Presiona en Editar
2. Se muestra una interfaz con los campos: nombre, iniciales, capacidad,

descripción, activo y categoría. Seguido por los botones Aceptar y cancelar.

3. Se llenan los campos
4. Presiona botón Siguiente
5. Se muestra una interfaz para seleccionar la estructura padre.
6. Se selecciona
7. Presiona el botón aceptar.
8. Se muestra una notificación con el mensaje: "Estructura editada".

Ver detalles

1. Presiona en el botón detalles
2. Se muestra una ventana con los detalles de la estructura

Buscar

1. Llena el campo de búsqueda
2. Se filtran los datos de la tabla.

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H15_PA15

Historia de Usuario: 15

Nombre: Configurar estructuras

Descripción: Prueba de la funcionalidad de configurar estructuras

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona distribución.
2. Presiona estructuras.
3. Se carga la vista con una lista de estructuras de forma jerárquica.
9. Presiona en configurar
10. Se muestra una interfaz con los campos: Subdirector, Especialista de complejo, técnico general y centro de costo. Seguido por los botones aceptar y cancelar.
11. Se llenan los campos
12. Presiona botón Aceptar
13. Se muestra una notificación con el mensaje: "Estructura configurada".

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H16_PA16

Historia de Usuario: 16

Nombre: Gestionar eventos

Descripción: Prueba de las funcionalidades adicionar, eliminar, modificar y buscar eventos

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona distribución.
2. Presiona eventos.

3. Se carga la vista con una lista de eventos.

Adicionar:

1. Presiona en Adicionar
2. Se muestra una interfaz con los campos: nombre, descripción, horario, activo, evento padre y clasificación. Seguido por los botones aceptar y cancelar.
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Evento creado".

Eliminar

1. Presiona el botón Eliminar
2. Se muestra una interfaz con el mensaje: "¿Está seguro que desea eliminar este evento?"
3. Presiona el botón Aceptar
4. Se muestra una notificación con el mensaje "Evento eliminado".

Modificar

1. Presiona en Editar
2. Se muestra una interfaz con los campos: nombre, descripción, horario, activo, evento padre y clasificación. Seguido por los botones aceptar y cancelar
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje "Evento editado".

Buscar

1. Llena el campo de búsqueda
2. Se filtra la lista

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H17_PA17

Historia de Usuario: 17

Nombre: Gestionar horarios

Descripción: Prueba de las funcionalidades adicionar, eliminar, modificar y buscar horarios

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona distribución.
2. Presiona horarios.
3. Se carga la vista con una lista de horarios.

Adicionar:

1. Presiona en Adicionar
2. Se muestran los campos: nombre, activo, hora inicio, hora fin y días. Seguido por los botones aceptar y cancelar.
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Horario creado"

Eliminar

1. Presiona el botón Eliminar

2. Se muestra una interfaz con el siguiente mensaje: ¿Está seguro que desea eliminar este horario?
3. Presiona el botón Aceptar
4. Se muestra una notificación con el mensaje: "Horario eliminado".

Modificar

1. Presiona en Editar
2. Se muestran los campos: nombre, activo, hora inicio, hora fin y días. Seguido por los botones aceptar y cancelar
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Horario editado".

Buscar

1. Llena el campo de búsqueda
2. Se filtran los datos de la tabla

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H18_PA18

Historia de Usuario: 18

Nombre: Gestionar tipos de áreas

Descripción: Prueba de las funcionalidades adicionar, eliminar, modificar y buscar tipos de áreas

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona configuración.
2. Presiona configuración de tipos de área.
3. Se carga la vista con una lista de tipos de área.

Adicionar:

1. Presiona en Adicionar
2. Se muestra una interfaz con los campos: nombre, descripción, activo y los botones aceptar y cancelar.
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Tipo de área creada".

Eliminar

1. Presiona el botón Eliminar
2. Se muestra una interfaz con el siguiente mensaje: ¿Está seguro que desea eliminar este tipo de área?.
3. Presiona el botón Aceptar
4. Se muestra una notificación con el mensaje: "Tipo de área eliminada".

Modificar

1. Presiona en Editar
2. Se muestra una interfaz con los campos: nombre, descripción, activo y los botones aceptar y cancelar
3. Se llenan los campos
4. Presiona botón Aceptar
5. Se muestra una notificación con el mensaje: "Tipo de área editada"

<p>Buscar</p> <ol style="list-style-type: none"> 1. Llena el campo de búsqueda 2. Se filtran los datos de la tabla
<p>Resultado: Satisfactorio.</p>

Caso de prueba de aceptación	
Código:H19_PA19	Historia de Usuario: 19
Nombre: Administrar período de reservación	
Descripción: Prueba de las funcionalidades adicionar y editar período de reservación	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe estar autenticado • El usuario debe encontrarse en el módulo de configuración 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Presiona reservación. 2. Presiona configuración de reservación. 3. Se carga la vista con una lista de períodos de reservación. <p>Adicionar:</p> <ol style="list-style-type: none"> 1. Presiona en Adicionar configuración por defecto <p>Modificar</p> <ol style="list-style-type: none"> 1. Presiona en Editar 2. Se muestra una interfaz con el campo período de reservación y el botón editar. 3. Se llenan los campos 	

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

1. Presiona reservación.
2. Presiona configuración de reservación.
3. Se carga la vista con una lista de períodos de reservación.

Adicionar:

1. Presiona en Adicionar configuración por defecto

Modificar

1. Presiona en Editar
2. Se muestra una interfaz con el campo período de reservación y el botón editar.
3. Se llenan los campos

4. Presiona botón Editar.

5. Se muestra una notificación con el siguiente mensaje: " Cantidad editada".

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H20_PA20

Historia de Usuario: 20

Nombre: Administrar elementos a mostrar

Descripción: Prueba de las funcionalidades adicionar y editar elementos a mostrar

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona reservación.
2. Presiona configuración de reservación.
3. Presiona elementos a mostrar
4. Se carga la vista con una lista de elementos a mostrar.

Adicionar:

1. Presiona en Adicionar configuración por defecto

Modificar reservación por platos

1. Presiona en Editar
2. Se muestra una interfaz con un campo para editar la cantidad de reservaciones por plato. Seguido por el botón Editar.

3. Se llenan los campos
4. Presiona botón Editar
5. Se muestra una notificación con el mensaje: "Cantidad editada".

Modificar reservación por calendario

1. Presiona en Editar
2. Se muestra una interfaz con un campo para editar la cantidad de reservaciones por calendario. Seguido por el botón Editar.
3. Se llenan los campos
4. Presiona botón Editar
5. Se muestra una notificación con el mensaje: "Cantidad editada".

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H21_PA21

Historia de Usuario: 21

Nombre: Administrar datos de contacto.

Descripción: Prueba de las funcionalidades adicionar y editar datos de contacto

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona reservación.
2. Presiona configuración de reservación.

3. Presiona datos de contacto

4. Se carga la vista con una lista de datos de contacto.

Adicionar:

1. Presiona en Adicionar datos de contacto por defecto

Modificar

1. Presiona en Editar

2. Se muestra los campos: Teléfono, correo y dirección. Seguido por el botón Editar

3. Se llenan los campos

4. Presiona botón Editar

5. Se muestra una notificación con el mensaje: "Datos de contacto editados".

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H22_PA22

Historia de Usuario: 22

Nombre: Administrar responsables de área.

Descripción: Prueba de las funcionalidades adicionar y eliminar responsables de área y personas a área

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona reservación.

2. Presiona asignar responsable de reservación.

3. Se carga la vista con una lista de áreas.

Adicionar responsable:

1. Presiona en Adicionar responsable

2. Se muestra una lista de responsables asignados, seguido por una lista de personas con el botón asignar.

3. Presiona el botón asignar

4. Se muestra una notificación con el mensaje : “Responsable asociado”

Eliminar responsable:

1. Presiona asignar responsable

2. En la lista de responsables presiona el botón eliminar.

3. Se muestra una notificación con el mensaje: “Responsable eliminado”.

Adicionar personas:

1. Presiona en Adicionar personas

2. Se muestra una lista de responsables, seguida por una lista de personas con el botón asignar. Seguida por una lista de personas ya asignadas.

3. Adiciona una persona a área.

4. Se muestra una notificación con el mensaje : “Responsable asociado”.

Eliminar persona asociada:

1. Presiona asignar personas

2. En la lista de personas asociadas presiona el botón eliminar.

3. Se muestra una notificación con el mensaje: “Persona eliminada”.

Resultado: Satisfactorio.

Caso de prueba de aceptación	
Código:H23_PA23	Historia de Usuario: 23
Nombre: Gestionar módulos por institución.	
Descripción: Prueba de las funcionalidades adicionar y eliminar módulos a una institución	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe estar autenticado • El usuario debe encontrarse en el módulo de configuración 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Presiona seguridad. 2. Presiona instituciones. 3. Se carga la vista con una lista de instituciones. 4. Presiona Editar 5. Se muestra una interfaz con los campos nombre, activo, descripción y módulos. Seguido por los botones Aceptar y Cancelar. 6. Selecciona los módulos. 7. Presiona Aceptar 8. Se muestra una notificación con el siguiente mensaje: "Institución editada" 	
Resultado: Satisfactorio.	

Caso de prueba de aceptación	
Código:H24_PA24	Historia de Usuario: 24
Nombre: Gestionar instituciones.	

Descripción: Prueba de las funcionalidades adicionar, editar y eliminar instituciones

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona seguridad.
2. Presiona instituciones.
3. Se carga la vista con una lista de instituciones.

Adicionar

1. Presiona adicionar
2. Se muestra una interfaz con los campos nombre, activo, descripción y módulos. Seguido por los botones Aceptar y Cancelar.
3. Llena los campos
4. Presiona Aceptar
5. Se muestra una notificación con el siguiente mensaje: "Institución creada"

Editar

1. Presiona editar
2. Se muestra una interfaz con los campos nombre, activo, descripción y módulos. Seguido por los botones Aceptar y Cancelar
3. Llena los campos
4. Presiona Aceptar
5. Se muestra una notificación con el siguiente mensaje: "Institución editada".

Eliminar

1. Presiona eliminar
2. Se muestra una interfaz con el siguiente mensaje: "¿Está seguro que desea eliminar esta institución?"
3. Presiona aceptar
4. Se muestra una notificación con el siguiente mensaje: "Institución eliminada"

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H25_PA25

Historia de Usuario: 25

Nombre: Administrar configuraciones del proceso

Descripción: Prueba de las funcionalidades habilitar o deshabilitar las configuraciones del proceso.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona reservación.
2. Presiona configuración del proceso de reservación.
Adicionar
3. Presiona adicionar configuración por defecto
Habilitar
1. Presiona habilitar
2. Se muestra una notificación con el mensaje: "Proceso de reservación cambiado"

exitosamente”.

Deshabilitar

1. Presiona deshabilitar
2. Se muestra una notificación con el mensaje: “Proceso de reservación cambiado exitosamente”.

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H26_PA26

Historia de Usuario: 26

Nombre: Gestionar configuración de cobro.

Descripción: Prueba de las funcionalidades adicionar, activar, agregar valores y desactivar una configuración de cobro

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona facturación.
2. Presiona configuración de cobro.
3. Se carga la vista con una lista de configuraciones del proceso de reservación.

Adicionar

1. Presiona adicionar
2. Se añade una nueva configuración de cobro por defecto.

Agregar valores

1. Presiona agregar valores
2. Se muestra una interfaz con los campos: nombre y descripción con los botones aceptar y cancelar. Seguido por los campos categoría, residente, tipo de cobro, evento, precio y el botón asociar. Por último se muestra una lista con las configuraciones agregadas con anterioridad.
3. Llena los campos
4. Presiona Aceptar
5. Se muestra una notificación con el siguiente mensaje: “Valores de la configuración de cobro agregados”

Habilitar

1. Presiona habilitar
2. Se muestra una notificación con el mensaje: “Configuración activada”

Deshabilitar

1. Presiona deshabilitar

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H27_PA27

Historia de Usuario: 27

Nombre: Administrar configuración de comensales.

Descripción: Prueba de las funcionalidades adicionar, activar, editar y desactivar una configuración de cobro

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona configuración.
2. Presiona configuración de comensales.
3. Se carga la vista con una lista de configuraciones del proceso de reservación.

Adicionar

1. Presiona adicionar por defecto
2. Se añade una configuración por defecto

Adicionar personas a la configuración

1. Presiona adicionar personas
2. Se muestra una lista de personas
3. Presiona adicionar
4. Se muestra una notificación con el mensaje: "Comensal asignado"

Habilitar

1. Presiona habilitar
2. Se muestra una notificación con el siguiente mensaje: "Estado de la configuración cambiado"

Deshabilitar

1. Presiona deshabilitar
2. Se muestra una notificación con el siguiente mensaje: "Estado de la configuración cambiado"

Agregar valores

1. Presiona agregar valores
2. Se muestra una interfaz con los campos: nombre, categoría, residentes, áreas y el botón asociar.
3. Llena los campos
4. Presiona Asociar
5. Se muestra una notificación con el siguiente mensaje: "Configuración asociada"
6. Se registra la configuración en la lista de configuraciones.

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H28_PA28

Historia de Usuario: 28

Nombre: Administrar comensales.

Descripción: Prueba de funcionalidad de eliminar comensales activos

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona configuración.
 2. Presiona comensales.
 3. Se carga la vista con una lista de comensales activos
- Eliminar
1. Presiona eliminar

2. Se muestra una notificación con el mensaje “Comensal desactivado”.

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H29_PA29

Historia de Usuario: 29

Nombre: Gestionar permisos.

Descripción: Prueba de funcionalidad de añadir y eliminar permisos.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona seguridad.

2. Presiona roles y permisos.

3. Presiona editar

Adicionar

1. Se muestra una interfaz con una lista de permisos disponibles en la parte izquierda de la pantalla y en la parte derecha una lista con los permisos asociados

2. Presiona el permiso en la lista de permisos disponibles

3. Presiona añadir

4. Se añade el permiso a la lista de permisos otorgados.

Eliminar

1. Presiona el permiso de la lista de permisos otorgados

2. Presiona eliminar
3. Se pasa el permiso a la lista de permisos disponibles.

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H30_PA30

Historia de Usuario: 30

Nombre: Gestionar roles.

Descripción: Prueba de funcionalidad de añadir, editar y eliminar roles.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona seguridad.
2. Presiona roles y permisos.
 - Adicionar
 - 1. Se muestran los campos nombre y los permisos. Seguido por los botones Aceptar y Cancelar.
 - 2. Se llenan los campos y se seleccionan los permisos
 - 3. Presiona aceptar
 - 4. Se muestra una notificación con el mensaje: "Rol creado"
 - Editar
 - 1. Presiona el botón editar

2. Se muestran los campos nombre y los permisos. Seguido por los botones Aceptar y Cancelar. Se llenan los campos
 3. Presiona aceptar
 4. Se muestra una notificación con el mensaje: "Rol editado"
- Eliminar
1. Presiona eliminar
 2. Se muestra el mensaje: ¿Está seguro que desea eliminar este rol?
 3. Presiona aceptar
 4. Se muestra una notificación con el mensaje: "Rol eliminado"

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H31_PA31

Historia de Usuario: 31

Nombre: Gestionar usuarios.

Descripción: Prueba de funcionalidad de añadir, editar y eliminar usuarios.

Condiciones de ejecución:

- El usuario debe estar autenticado
- El usuario debe encontrarse en el módulo de configuración

Pasos de ejecución:

1. Presiona seguridad.
 2. Presiona usuarios.
- Adicionar

1. Presiona adicionar
2. Se muestra una interfaz con los campos: nombre de usuario, correo electrónico, institución, rol, contraseña y confirmar contraseña. Seguido por los botones aceptar y cancelar.
3. Se llenan los campos
4. Presiona aceptar
5. Se muestra una notificación con el mensaje: "Usuario creado".

Editar

1. Presiona editar
2. Se muestra una interfaz con los campos: nombre de usuario, correo electrónico, institución, rol, contraseña y confirmar contraseña. Seguido por los botones aceptar y cancelar
3. Se llenan los campos
4. Presiona aceptar
5. Se muestra una notificación con el mensaje: "Usuario editado".

Eliminar

1. Presiona eliminar
2. Se muestra una interfaz con el mensaje: "¿Está seguro que desea eliminar este usuario?"
3. Presiona aceptar
4. Se muestra una notificación con el mensaje: "Usuario eliminado".

Resultado: Satisfactorio.

Caso de prueba de aceptación

Código:H31_PA32	Historia de Usuario: 32
Nombre: Autenticación de usuario	
Descripción: Prueba de funcionalidad de autenticar usuario.	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe introducir en el navegador la dirección del sitio.	
Pasos de ejecución: <ol style="list-style-type: none">1. Se muestra una interfaz con los campos usuario y contraseña y el botón Entrar.2. Llena los campos.3. Presiona Entrar.4. Se muestra la interfaz principal del sistema con la notificación "Usuario logueado".	
Resultado: Satisfactorio.	

Anexo 5: Aval de satisfacción de cliente. Fuente: Dirección de Informatización.



La Habana, 26 de Octubre de 2023

"Año 65 de la Revolución"

Dirección de Informatización de la Universidad de las Ciencias Informáticas.

Acta de Conformidad

Por este medio se comunica la satisfacción de la Dirección de Informatización de la Universidad de las Ciencias Informáticas con el Módulo de Configuración para el Sistema de Gestión de Alimentación en su versión 2.0 entregado por Alejandro Javier Molina Santos y Enrique Ferreiro Ávila. Se certifica que el producto cumplió con los 32 requisitos especificados y fue entregado en los plazos establecidos. El sistema fue probado y cumple con los estándares requeridos. Este módulo será utilizado en la versión 2.0 del sistema actual con el fin de comercializarlo y contribuir económicamente al progreso de la universidad. Por ello, el centro acepta el software en su estado actual y se compromete con la utilización del mismo en los términos y condiciones acordados, así como guiar la continuidad del desarrollo de las recomendaciones realizadas.

Cordialmente,



A handwritten signature in black ink, appearing to read 'Yosbel Falero Vento', written over a horizontal line.

Ing. Yosbel Falero Vento

Director de Informatización

Universidad de las Ciencias Informáticas