



FACULTAD DE TECNOLOGÍAS INTERACTIVAS

Sistema de gestión de información de seguridad y protección para la Empresa de Seguridad y Protección de la Aviación Civil

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Zamir Alarcón Cabrera

Tutores: Lic. Ernesto Guerra Arias
M. Sc. Yadira Ramírez Rodríguez

La Habana, diciembre de 2024

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título “***Sistema de gestión de información de seguridad y protección para la Empresa de Seguridad y Protección de la Aviación Civil***”, concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firman la presente a los 3 días del mes de diciembre del año 2024.

Zamir Alarcón Cabrera



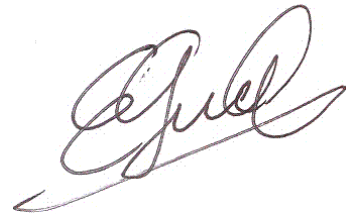
Firma del Autor

Yadira Ramírez Rodríguez



Firma del Tutor

Ernesto Guerra Arias



Firma del Tutor

DATOS DE CONTACTO

Yadira Ramírez Rodríguez. Graduada en el año 2007 de La Universidad de las Ciencias Informáticas (UCI). Posee categoría docente principal de Profesor Auxiliar y es Máster en Calidad de Software. Ha impartido asignaturas de las disciplinas de Práctica Profesional e Ingeniería y Gestión de Software. Actualmente es jefa del Departamento Docente de Informática de la Facultad 4 de la UCI. Sus líneas de investigación: Ingeniería de Software, Calidad de Software y Gestión de Proyectos.

Ernesto Guerra Arias. Graduado en el año 1991 de la Facultad de Matemática y Ciencias de la Computación de la Universidad de La Habana. Actualmente ocupa el cargo de Director de Informática y Comunicaciones en ESPAC S.A y es licenciado en matemáticas. Posee una amplia experiencia en roles de investigación y especialización en informática, destacándose como investigador en BioCen y especialista en diversas entidades relacionadas con la aviación civil cubana. Desde 2016, ha dirigido el departamento de informática y comunicaciones en ESPAC S.A. Ha contribuido al ámbito académico con cursos y asesorías en estadística, procesamiento automatizado de ensayos y sistemas de indicadores económicos.

RESUMEN

La gestión aeroportuaria es esencial para el funcionamiento eficiente de los aeropuertos, y la Empresa de Seguridad y Protección de la Aviación Civil en Cuba desempeña un papel crucial en la provisión de servicios de seguridad y protección en todos los aeropuertos del país. Actualmente, la empresa utiliza una aplicación web para gestionar su información. Sin embargo, esta aplicación, en uso desde 2011, no se adapta a los nuevos requerimientos de información. Para paliar estas deficiencias, se han utilizado hojas de cálculo de Microsoft Excel, lo que ha introducido problemas críticos debido a la complejidad y al riesgo de errores humanos. Este documento aborda el problema de cómo reemplazar el sistema web limitado y las hojas de cálculo empleadas actualmente para contribuir a una gestión más eficiente de la información en la Empresa de Seguridad y Protección de la Aviación Civil. A través de métodos y técnicas de investigación, se identificaron los mecanismos necesarios para desarrollar un sistema web que cubra las necesidades de la empresa. El desarrollo del nuevo sistema se realizó utilizando React como framework y se guió por la metodología de desarrollo de software Programación Extrema (XP), enfocada en mejorar la calidad del software y la capacidad de respuesta a los cambios en los requisitos del cliente.

Palabras clave

Empresa de Seguridad y Protección de la Aviación Civil de Cuba, Gestión de Información, Gestión de Incidencias

ABSTRACT

Airport management is essential for the efficient operation of airports, and the Civil Aviation Security and Protection Company in Cuba plays a crucial role in providing security and protection services at all airports in the country. Currently, the company uses a web application to manage its information. However, this application, in use since 2011, does not meet the new information requirements. To address these deficiencies, Microsoft Excel spreadsheets have been used, which has introduced critical problems due to complexity and the risk of human errors. This document addresses the problem of how to replace the limited web system and spreadsheets currently used to contribute to more efficient information management in the Civil

Aviation Security and Protection Company. Through research methods and techniques, the necessary mechanisms were identified to develop a web system that meets the company's needs. The development of the new system was carried out using React as a framework and guided by the Extreme Programming (XP) software development methodology, focused on improving software quality and responsiveness to changes in client requirements.

Keywords

Civil Aviation Security and Protection Company, Information Management, Incident Management

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LA GESTION DE INFORMACIÓN DE SEGURIDAD Y PROTECCIÓN EN LOS AEROPUERTOS CUBANOS.....	6
I.1 Fundamentos teóricos sobre la gestión de información de seguridad y protección	6
I.1.1 Procesos y sistemas implicados en la gestión de información de seguridad y protección de la empresa ESPAC.	7
I.2 Gestión de información de seguridad y protección en la empresa ESPAC	8
I.3 Análisis de sistemas homólogos	9
I.3.1 Sistemas homólogos en el ámbito internacional	9
I.3.2 Sistemas homólogos en el ámbito nacional	10
I.3.3 Resultados del estudio de sistemas homólogos	11
I.4 Herramientas y tecnologías utilizadas.....	12
I.4.1 Lenguaje de programación	12
I.4.2 Framework de desarrollo	13
I.4.3 Entorno de programación	16
I.4.4 Metodología de desarrollo de software	17
Conclusiones del capítulo.....	20
CAPÍTULO II: DISEÑO Y CARACTERISTICAS DEL SISTEMA DE GESTIÓN DE INFORMACIÓN COMO SOLUCIÓN AL PROBLEMA PLANTEADO	21
II.1 Presentación de la solución.....	21
II.2 Historias de usuarios	25
II.3 Requisitos no funcionales.....	31
RnF1-Software	31
RnF2-Hardware	31
RnF3-Interfáz o apariencia externa	31
RnF4-Rendimiento	31
II.4 Planificación de las entregas	32
Estimación de esfuerzo por historia de usuario	32
Plan de Iteraciones	33
Plan de entregas	34
II.5 Diseño del Sistema.....	34
II.5.1 Patrón de arquitectura	34
II.5.2 Patrones de diseño	36
Conclusiones del capítulo.....	38
CAPÍTULO III: IMPLEMENTACION Y VALIDACION DEL SISTEMA GISPV4	40
III.1 Codificación	40

Tareas de ingeniería	40
III.2 Pruebas	44
III.2.1 Pruebas unitarias	45
III.2.2 Resultados de las pruebas unitarias.....	49
III.2.3 Pruebas de aceptación.....	50
III.3.4 Análisis de las pruebas de aceptación	55
Conclusiones del capítulo.....	56
CONCLUSIONES FINALES	57
RECOMENDACIONES	58
REFERENCIAS BIBLIOGRÁFICAS.....	59
ANEXOS	62

ÍNDICE DE TABLAS

Tabla 1. Sistemas homólogos en el ámbito nacional e internacional.....	10
Tabla 2. Frameworks de desarrollo web más utilizados	14
Tabla 3. Metodologías de desarrollo de software más utilizadas	18
Tabla 4. Historia de usuario #2. Gestionar unidades	26
Tabla 5. Historia de usuario #3 Gestionar Instalaciones	27
Tabla 6. Historia de usuario #12 Adicionar incidencia	29
Tabla 7. Estimación de esfuerzo por historia de usuario	32
Tabla 8. Plan de iteraciones	33
Tabla 9. Plan de entregas	34
Tabla 10. Tarea de ingeniería 1	41
Tabla 11. Tarea de ingeniería 2	41
Tabla 12. Tarea de ingeniería 8	41
Tabla 13. Tarea de ingeniería 9	42
Tabla 14. Tarea de ingeniería 20	42
Tabla 15. tarea de ingeniería 21	43
Tabla 16. Tarea de ingeniería 24	43
Tabla 17. Tarea de ingeniería 26	44
Tabla 18. Historia de usuario # 4. Gestionar posiciones	62
Tabla 19. Historia de usuario #1. Iniciar sesión	63
Tabla 20. Historia de usuario #5. Gestionar grupos	64
Tabla 21. Historia de usuario #6. Gestionar agentes	65
Tabla 22. Historia de usuario #7. Gestionar medios	66
Tabla 23. Historia de usuario #8. Gestionar configuración de medios	67
Tabla 24. Historia de usuario #9. Listar posiciones en el mapa	68
Tabla 25. Historia de usuario #10. Actualizar fuerzas de posición	69
Tabla 26. Historia de usuario #11. Gestionar medios disponibles	70
Tabla 27. Historia de usuario #13. Listar incidencias	71
Tabla 28. Historia de usuario #14. Generar gráfico de reportes de incidencias	71

ÍNDICE DE FIGURAS

Figura 1. Solución. Mapa²³

Figura 2. Propuesta de solución²⁴

Figura 3. Representación arquitectónica³⁶

Figura 4. Pruebas unitarias al gestionar unidades UEB (Fuente: elaboración propia)⁴⁶

Figura 5. Pruebas unitarias al gestionar instalaciones (Fuente: elaboración propia)⁴⁷

Figura 6. Pruebas unitarias al gestionar posiciones (Fuente: elaboración propia)⁴⁸

Figura 7. Resultado de las pruebas unitarias⁵⁰

Figura 8. Resultados de las pruebas de aceptación. (Fuente: Elaboración propia)⁵⁶

AVAL DEL CLIENTE

 **ESPAC**
EMPRESA DE SEGURIDAD Y PROTECCIÓN DE LA AVIACIÓN CIVIL

La Habana 27 de noviembre de 2024
Año 66 de la Revolución

ACTA DE CONFORMIDAD

Por este medio comunicamos nuestra satisfacción con el sistema de gestión de información de seguridad y protección entregado por Zanir Alarcón Cabrera. Certificamos que el producto entregado cumplió con la totalidad de los requisitos especificados y fue entregado en los plazos establecidos. El sistema fue probado y cumple con los estándares requeridos. La Empresa de Seguridad y Protección de la Aviación Civil acepta el software en su estado actual y se compromete con la utilización del mismo en los términos y condiciones acordados.

Y PARA QUE CONSTE la aceptación del trabajo realizado, se firma la presente ACTA cordialmente

Leonardo Sarmiento Izquierdo
Director de Seguridad y Protección
ESPAC S.A.



Dirección: Carretera a Aerocaribbean, Km 1 ¼ Wajay, Boyeros, La Habana

INTRODUCCIÓN

Con la generalización del uso del transporte aéreo, concreción física real de la globalización, los aeropuertos han pasado de ser simples chalets destinados a proteger a los viajeros de las inclemencias del tiempo, a ser complejas infraestructuras vanguardistas, dotadas de sofisticadas tecnologías mecánicas y de telecomunicaciones, de áreas de comercio, ocio y restauración. Cualquier actor del transporte aéreo, según referencia general de OACI (Organización de Aviación Civil Internacional), debe prestar unos servicios seguros, regulares y eficientes. Es decir, los aeropuertos tendrán que lograr una alta eficiencia en sus referencias operacionales y de gestión, en definitiva, estar basados en modelos de calidad(Andrada & Santos, 2019).

La gestión aeroportuaria es un componente vital en el funcionamiento eficiente de cualquier aeropuerto. Un gestor aeroportuario desempeña un papel clave en la supervisión y coordinación de diversas actividades para garantizar que las operaciones transcurran sin contratiempos. Desde la planificación estratégica hasta la implementación de medidas de seguridad, su papel abarca una amplia gama de responsabilidades.(Roberto, 2023)

La Empresa de Seguridad y Protección de la Aviación Civil (ESPAC) en Cuba desempeña un papel fundamental en la provisión de servicios de seguridad y protección para la aviación civil en el país. Operando en todos los aeropuertos del país y formando parte de la Corporación de la Aviación Cubana S.A. (CACSA), ESPAC se dedica a garantizar la seguridad en los aeropuertos, proteger a los pasajeros y la tripulación, y asegurar que las operaciones aéreas se realicen de manera segura y eficiente.

La correcta gestión de la información por parte de ESPAC es crucial para las operaciones del aeropuerto por varias razones. En primer lugar, la información precisa y actualizada es esencial para identificar y mitigar amenazas potenciales, incluyendo la gestión de datos sobre pasajeros, equipaje y carga, así como la coordinación con otras agencias de seguridad. Además, una buena gestión de la información permite una coordinación eficiente entre los diferentes departamentos del aeropuerto, como el control de tráfico aéreo, la gestión de terminales y los servicios de tierra, lo que ayuda a minimizar retrasos y a mejorar la experiencia del pasajero.

Asimismo, la aviación civil está altamente regulada, y la correcta gestión de la información asegura que el aeropuerto cumpla con todas las normativas locales e internacionales, evitando sanciones y garantizando la seguridad de las operaciones. En caso de una emergencia, tener acceso rápido y preciso a la información puede ser la diferencia entre una respuesta efectiva y un desastre, incluyendo planes de evacuación, datos de contacto de emergencia y protocolos de seguridad. Por todas estas razones, la gestión de la información en ESPAC es un componente esencial para el éxito y la seguridad de las operaciones aeroportuarias en Cuba. La Empresa de Seguridad y Protección de la Aviación Civil (ESPAC) organiza sus servicios en los aeropuertos por posiciones estratégicas donde se ubican agentes de seguridad y protección para garantizar la protección de cada instalación. En estos momentos se cuenta con una aplicación web (GISP) mediante la cual se gestiona la información general de cada turno de trabajo y las incidencias de seguridad que ocurren.

Desde su implementación en 2011, y tras varias actualizaciones, GISP ha alcanzado su versión número 3. Sin embargo, esta aplicación no se adapta a los nuevos requerimientos de información que han surgido y carece de capacidades de gestión en tiempo real. Esto significa que cualquier cambio o actualización realizada durante el turno no se reflejará hasta que se complete el registro al final del mismo. Esta limitación puede resultar en decisiones basadas en información desactualizada o incorrecta, afectando negativamente la seguridad y la eficiencia del aeropuerto. Otra carencia de la aplicación es la falta de la información de la localización geográfica de cada posición donde se brindan los servicios, lo que no permite la representación en un mapa del estado real de la seguridad de las instalaciones.

Para suplir algunas de estas carencias se diseñaron hojas de cálculo de Microsoft Excel, lo que presenta varios problemas críticos al contener numerosas fórmulas complejas y hojas interrelacionadas. Esta complejidad aumenta el riesgo de errores humanos, ya que un simple error de escritura puede tener consecuencias graves. Un error en una fórmula o en la entrada de datos puede propagarse a través de las hojas interrelacionadas, llevando a la pérdida total de información crítica o a la generación de datos incorrectos.

Además, la gestión de información en Excel carece de las capacidades avanzadas de seguridad y control de versiones que ofrecen otros sistemas más robustos. Esto significa que

la información puede ser fácilmente alterada o perdida sin posibilidad de recuperación, lo que compromete la seguridad y la eficiencia de las operaciones aeroportuarias.

Debido a esta **situación problemática** se plantea como **problema** de investigación: ¿Cómo mejorar la gestión de información de seguridad y protección en la empresa ESPAC?

Se define como **objeto de estudio** en esta investigación la gestión de la información en las operaciones de seguridad y protección en la aviación civil. Se enmarca como **campo de acción**: La gestión de la información en las operaciones de seguridad y protección de la aviación civil en los aeropuertos cubanos, con un enfoque particular en la gestión de incidencias y el control de posiciones.

Para dar solución al problema planteado se define como **objetivo general**: Desarrollar un sistema de gestión de información de seguridad y protección para la empresa ESPAC.

Tareas de investigación:

1. Análisis de un marco teórico a través del estudio del estado actual del arte para establecer una base sólida y actualizada sobre la gestión de información.
2. Investigación de las necesidades específicas de ESPAC en cuanto a la gestión de información de seguridad y protección para identificar áreas de mejora y requerimientos particulares.
3. Implementación de un sistema web para la gestión de información de seguridad y protección para optimizar los procesos y asegurar una mejor organización de los datos.
4. Verificación y validación de la solución propuesta a través de pruebas exhaustivas de software para garantizar su efectividad y fiabilidad en el entorno operativo.

Métodos y técnicas de investigación

El método y la técnica se consideran métodos orientados para lograr un fin y hacen referencia a los procedimientos para lograr el mismo. Se entiende por método, al conjunto de pasos ordenados y secuencias los cuales se siguen para obtener un resultado. Por otro lado, la técnica hace referencia a la manera en que se aplica el método y por tal motivo se considera un procedimiento más específico(Alfonso et al., 2020).

Métodos Teóricos:

Histórico-Lógico: El método histórico-lógico se utilizó para analizar y comprender la evolución y desarrollo de los procesos que influyen en la gestión de información en ESPAC. Se analizaron los cambios y mejoras implementadas en diferentes períodos, así como las condiciones que influyeron en estos cambios.

Análisis sintético: se utilizó para extraer e identificar conceptos, características y otros elementos de la bibliografía consultada que posteriormente ayudaron a establecer una propuesta adecuada a las necesidades del sistema. Se descompuso el problema de la gestión de la información en sus componentes básicos: Gestión de agentes de seguridad y protección, gestión de incidencias y gestión de medios básicos para el cumplimiento de las operaciones aeroportuarias. Se analizaron las características, ventajas y limitaciones de las tecnologías actuales utilizadas por ESPAC en el proceso de gestión de información y se identificaron las necesidades específicas de la empresa en términos de recopilación, almacenamiento y análisis de datos en tiempo real. Finalmente se integraron los hallazgos del análisis para formar una visión completa de las deficiencias y oportunidades de mejora en la gestión de la información.

Métodos Empíricos:

Observación: se emplea este método para conocer la esencia de la problemática definida, que sirvió de base para el planteamiento del problema de la investigación. Se Observó de manera directa y participativa los procesos de gestión del personal y control de posiciones en, los cuales se gestionan a través de hojas de cálculo de Excel y el almacenamiento y análisis de incidencias gestionados por el sistema GISP.

Entrevista: Se le realizó al técnico de seguridad y protección de la Unidad Empresarial de Base (UEB) José Martí de ESPAC. La entrevista tuvo como objetivo obtener una visión detallada de los procesos actuales y las limitaciones tecnológicas que enfrenta la empresa.

Estructura del documento:

El presente documento está conformado por 3 capítulos

Capítulo 1: Fundamentos y referentes teórico-metodológicos sobre la gestión de información de seguridad y protección en los aeropuertos cubanos.

Este capítulo se centra en el estudio del estado actual del arte, aspectos teóricos y conceptos fundamentales sobre la gestión de información de seguridad y protección de ESPAC. Se

analizan las soluciones existentes en el ámbito nacional e internacional. Se escoge una metodología de desarrollo de software para guiar el proceso, así como las tecnologías a utilizar.

Capítulo 2: Diseño y características del sistema de gestión de información como solución al problema planteado.

Este capítulo presenta la descripción del sistema GISPV4, sus características y requisitos funcionales y no funcionales partiendo de la metodología de software. Se definen los patrones de diseño y su arquitectura.

Capítulo 3: Implementación y validación del sistema de gestión de información.

Este capítulo se centra en la evaluación del sistema GISPV4 para asegurar que cumple con los requisitos establecidos y funciona correctamente en el entorno operativo de ESPAC. Se describen los métodos y procedimientos utilizados para validar y verificar el sistema, así como los resultados obtenidos.

CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LA GESTION DE INFORMACIÓN DE SEGURIDAD Y PROTECCIÓN EN LOS AEROPUERTOS CUBANOS.

La gestión de la información de seguridad y protección en los aeropuertos es un componente crítico para garantizar la seguridad de las operaciones aéreas y la protección de los pasajeros, el personal y las instalaciones. En el contexto de los aeropuertos cubanos, esta gestión adquiere una relevancia particular debido a la necesidad de mantener altos estándares de seguridad en un entorno dinámico y en constante evolución.

Este capítulo tiene como objetivo proporcionar una base teórica y metodológica sólida para entender los procesos y sistemas utilizados en la gestión de la información de seguridad y protección en los aeropuertos cubanos. Se explorarán los conceptos fundamentales, las teorías y los modelos que sustentan esta gestión, así como las soluciones tecnológicas existentes tanto a nivel nacional como internacional.

Además, se analizarán las metodologías de desarrollo de software más adecuadas para guiar el proceso de implementación de un nuevo sistema de gestión de información. Se seleccionarán las tecnologías más avanzadas y pertinentes para asegurar que el sistema propuesto no solo cumpla con los requisitos actuales, sino que también sea escalable y adaptable a futuras necesidades

I.1 Fundamentos teóricos sobre la gestión de información de seguridad y protección

El término gestión de la información se nombra tanto últimamente que nos puede parecer que es un invento reciente, pero nada más lejos de la realidad. Desde hace cientos de años las empresas, los líderes y los grandes estrategias políticos y económicos saben que la información es poder y que gestionar los datos, las noticias que tenemos, puede marcar la diferencia entre el éxito... y el fracaso.(Directivos, 2016).

La información por sí misma no resulta útil si no es la que realmente se necesita, o bien si no se puede o no se sabe tratar o procesar. La gestión de la información hace referencia al proceso por el que se obtienen, despliegan o utilizan recursos básicos para el manejo de información dentro y en favor de la sociedad a la que sirve(García, 2021).

Gracias a estos, las empresas pueden organizar y vincular los datos que generan en diferentes áreas o procesos. Así, sus trabajadores pueden localizar, en una sola plataforma, información de alto valor para sus labores en tiempo real o en cualquier momento que lo necesiten. Por ejemplo, para captar, recolectar, almacenar, analizar, manejar y diseminar datos (Camila, 2023).

I.1.1 Procesos y sistemas implicados en la gestión de información de seguridad y protección de la empresa ESPAC.

La gestión de la información de seguridad y protección en aeropuertos implica una serie de procesos y sistemas diseñados para garantizar la seguridad de las operaciones aéreas y la protección de los pasajeros, el personal y las instalaciones. A continuación, se presentan los conceptos fundamentales, teorías y modelos que sustentan esta gestión:

Gestión del personal de seguridad y protección

El trabajo de un agente de seguridad de aeropuerto es crucial para garantizar la seguridad de los pasajeros y el personal en los aeropuertos. Esta profesión requiere de habilidades y conocimientos específicos, así como de una gran responsabilidad y compromiso (Author, 2023).

Los agentes de seguridad y protección en un aeropuerto desempeñan una variedad de tareas críticas que incluyen vigilancia Continua en Postas o Garitas, protección de equipajes, control de accesos, monitoreo de cámaras de video vigilancia y control de operaciones directas a las aeronaves. Dado que estas tareas son diversas y requieren una atención constante, es crucial tener un buen control de la asignación de cada agente a las diferentes posiciones estratégicas. Una asignación adecuada asegura que todas las áreas críticas estén cubiertas y que los agentes puedan responder de manera eficiente a cualquier incidente. Además, permite una distribución equitativa de la carga de trabajo, lo que reduce la fatiga y mejora el rendimiento de los agentes.

Control de posiciones

Supervisión y gestión de las posiciones asignadas a los agentes de seguridad y protección para asegurar que todas las áreas críticas del aeropuerto estén vigiladas de manera efectiva. No se trata solo de desplegar personal, sino de asegurarse de que estén en el lugar adecuado en el momento preciso. La eficiencia es clave para garantizar la seguridad sin afectar la operación.

Registro de incidencias

Es fundamental contar con una base de datos centralizada donde se registren todas las incidencias. Esto permite un acceso rápido y eficiente a la información cuando sea necesario. Cada registro debe incluir detalles como la fecha, hora, ubicación, descripción del incidente, personas involucradas y nivel de relevancia. Las incidencias deben clasificarse según su naturaleza, como amenazas de bomba, altercados, objetos sospechosos, entre otros (CPD, 2023).

Un procedimiento de gestión de incidencias es un conjunto de acciones para anticipar, resolver y documentar eventos no planificados en una organización. Su objetivo es guiar a los profesionales a través de una situación inesperada y ayudarles a volver a la normalidad lo más rápido posible. En otras palabras, el seguimiento de incidencias es un plan utilizado para detectar, comunicar y solucionar un incidente. Incluye las herramientas y pasos necesarios para manejar emergencias con el menor daño posible (Zendesk, 2023).

I.2 Gestión de información de seguridad y protección en la empresa ESPAC

En la gestión de información de seguridad y protección en la empresa ESPAC, intervienen varios factores clave. Entre ellos se encuentra la asignación de turnos de vigilancia a los agentes de seguridad y protección, lo cual es fundamental para garantizar una cobertura adecuada en todas las áreas. Además, se lleva a cabo la recopilación de datos sobre las diferentes posiciones donde son asignados los agentes en cada Unidad Empresarial de Base (UEB) del país. Este proceso incluye el registro diario de incidencias, vuelos diarios, medios de seguridad necesarios en cada posición y su disponibilidad.

Uno de los principales procesos en la gestión de la información es la asignación de agentes a posiciones. Este proceso implica designar a cada agente en turno a una posición específica, teniendo en cuenta las vulnerabilidades más destacadas de cada área. La correcta asignación de agentes es crucial para mantener los aeropuertos totalmente protegidos, asegurando que cada punto vulnerable esté cubierto por personal capacitado y preparado para responder a cualquier eventualidad. Este enfoque estratégico permite optimizar los recursos humanos y garantizar un alto nivel de seguridad en todas las instalaciones aeroportuarias.

Un sistema de gestión de seguridad y protección en la empresa ESPAC, donde el operador pueda interactuar de manera intuitiva y segura al asignar agentes a las posiciones, registrar

incidencias e información de operaciones y medios básicos, contribuiría significativamente a la mejora en la gestión de la información. Este sistema reemplazaría las complejas hojas de Excel actualmente utilizadas para las asignaciones de agentes a posiciones, aumentando la velocidad de operación al registrar incidencias en tiempo real. La implementación de un sistema así no solo optimizaría los recursos humanos y materiales, sino que también garantizaría una respuesta más rápida y eficiente ante cualquier eventualidad, mejorando la seguridad y protección en todas las instalaciones.

I.3 Análisis de sistemas homólogos

El estudio de sistemas homólogos en el proceso de desarrollo de software implica analizar y comparar sistemas similares para identificar mejores prácticas, patrones de diseño y posibles mejoras. Este enfoque permite a los desarrolladores aprender de las experiencias previas y aplicar soluciones probadas a nuevos proyectos, lo que puede resultar en un desarrollo más eficiente y efectivo. Puede ayudar a optimizar el ciclo de vida del desarrollo de software (SDLC), que incluye varias fases como análisis de requisitos, diseño, desarrollo, pruebas, despliegue y mantenimiento (Pathak, 2022).

Ventajas del estudio de sistemas homólogos:

- **Identificación de patrones exitosos:** Permite reconocer soluciones efectivas que se pueden adaptar a nuevos proyectos.
- **Reducción de riesgos:** Al aprender de los errores y aciertos de sistemas similares, se pueden evitar problemas comunes.
- **Mejora de la eficiencia:** Implementar prácticas probadas puede acelerar el desarrollo y reducir costos.
- **Innovación:** Comparar diferentes enfoques puede inspirar nuevas ideas y mejoras.

I.3.1 Sistemas homólogos en el ámbito internacional

En el ámbito internacional se tuvo como referencia **Trackforce Valiant**, este software ofrece una plataforma integral para la gestión de seguridad, incluyendo la asignación de turnos, seguimiento de patrullas, y registro de incidentes en tiempo real. Es muy utilizado en la industria de la seguridad privada. Aunque es fácil de usar, puede requerir tiempo para la configuración inicial. **Silvertrac Software**, una solución basada en la web que permite a las empresas de seguridad gestionar sus operaciones diarias. Incluye funcionalidades para la

asignación de agentes, seguimiento de actividades y registro de incidentes. **Guardhouse**, este sistema de gestión de seguridad permite la planificación y asignación de turnos, así como el seguimiento de la asistencia y el rendimiento de los agentes. También facilita la comunicación entre el personal de seguridad, aunque requiere conexión constante a internet para su funcionamiento óptimo.

Los sistemas homólogos mencionados anteriormente, como Trackforce Valiant, Silvertrac Software y Guardhouse ofrecen funcionalidades muy similares que fueron consideradas para el desarrollo del proyecto actual como, características avanzadas de gestión de incidencias, monitoreo en tiempo real, y análisis de datos, lo que puede proporcionar una base sólida para mejorar la eficiencia y efectividad de la gestión de información en ESPAC.

I.3.2 Sistemas homólogos en el ámbito nacional

En el ámbito nacional, se ha decidido estudiar el sistema web actualmente en uso en la empresa ESPAC, conocido como GISPV3. Este sistema registra las incidencias detectadas durante toda una jornada laboral, permitiendo el registro de datos del infractor, del agente que detecta la incidencia y una descripción detallada de la misma.

El autor recomendó el estudio de este sistema en particular debido a que, al estar actualmente en uso por la empresa ESPAC, se facilita la obtención de experiencias y la identificación de errores conocidos. Esta situación no solo permite un análisis detallado de las deficiencias y limitaciones del sistema actual, sino que también destaca la necesidad de incorporar nuevas funcionalidades que mejoren la eficiencia y efectividad de la gestión de información.

Además, el uso continuo del sistema por parte de ESPAC proporciona una base de datos rica en información real y relevante, lo que permite realizar pruebas y ajustes más precisos durante el desarrollo del nuevo sistema.

Tabla 1. Sistemas homólogos en el ámbito nacional e internacional.

Parámetro	Trackforce Valiant	Silvertrac Software	Guardhouse	GISPV3
Tipo de Software Libre	No	No	No	No
Licencia de Pago	Sí	Sí	Sí	No
Impacto en el negocio	Alto	Alto	Alto	Alto

Escalabilidad	Sí	Sí	Sí	No
Flexibilidad	Sí	Sí	Sí	No

I.3.3 Resultados del estudio de sistemas homólogos

El análisis de sistemas homólogos tanto en el ámbito internacional como nacional ha proporcionado una visión integral de las soluciones disponibles para la gestión de seguridad y protección en entornos empresariales. A partir de este estudio, se han identificado varias conclusiones clave que justifican la necesidad de desarrollar un nuevo sistema para ESPAC:

Adecuación Tecnológica:

Los sistemas internacionales, como Trackforce Valiant, Silvertrac Software y Guardhouse, ofrecen funcionalidades avanzadas y características robustas de gestión de seguridad. Sin embargo, estos sistemas pueden presentar desafíos en términos de configuración inicial y requisitos de conectividad constante, además no son de software libre y requieren licencias de pago, lo que podría representar un desafío financiero y de flexibilidad para ESPAC. lo que podría no ser ideal para las necesidades específicas de ESPAC.

Especificidad de las Necesidades de ESPAC:

El sistema GISPV3, actualmente en uso por ESPAC, ha demostrado ser valioso en términos de recopilación de datos y registro de incidencias. No obstante, sus limitaciones en términos de escalabilidad y flexibilidad destacan la necesidad de un sistema más avanzado y adaptado a las necesidades crecientes de la empresa.

Mejora de la Eficiencia y Efectividad:

La implementación de un nuevo sistema, diseñado específicamente para ESPAC, puede incorporar funcionalidades mejoradas que no solo abordan las limitaciones actuales, sino que también optimizan la gestión de información y la eficiencia operativa. Características como el monitoreo en tiempo real, la gestión avanzada de incidencias y el análisis de datos serán cruciales para mejorar la operatividad y la toma de decisiones.

Aprovechamiento de Datos Existentes:

El uso continuo de GISPV3 ha generado una base de datos rica en información relevante. Al desarrollar un nuevo sistema, se puede aprovechar esta base de datos para realizar pruebas

más precisas y ajustes que aseguren la efectividad y fiabilidad del nuevo sistema desde el primer día de su implementación.

Necesidad de Desarrollar un Nuevo Sistema

Tomando en cuenta las conclusiones del estudio, queda claro que desarrollar un nuevo sistema específico para ESPAC es esencial por las siguientes razones:

- **Personalización:** Un sistema nuevo puede ser diseñado y personalizado para cumplir con los requisitos exactos de ESPAC, garantizando una solución a medida que maximice la eficiencia y efectividad de la gestión de información.
- **Integración de Funcionalidades Avanzadas:** La integración de características avanzadas como monitoreo en tiempo real, análisis de datos y gestión centralizada de incidencias permitirá una mejora significativa en las operaciones diarias y en la toma de decisiones estratégicas.
- **Optimización de Recursos:** Un sistema diseñado específicamente para ESPAC optimizará el uso de recursos, minimizando los tiempos de configuración y capacitación, y proporcionando una experiencia de usuario más fluida y eficiente.

I.4 Herramientas y tecnologías utilizadas

Las herramientas para el desarrollo de software son programas o aplicaciones que facilitan el trabajo de los desarrolladores tienen como finalidad disminuir el tiempo y el estrés de cada etapa del desarrollo de software además mejorar los resultados obtenidos y dar mejores propuestas al cliente. La selección de estas herramientas es proponen un factor clave en el desarrollo y está dada principalmente por los requerimientos del sistema. En la siguiente sección se definen las tecnologías y herramientas que se escogieron para la implementación de la solución.

I.4.1 Lenguaje de programación

Para el desarrollo de sistemas web, existen varios lenguajes de programación que se destacan por su versatilidad, rendimiento y facilidad de uso como son JavaScript, Python, Java, PHP, C#, entre otros. Para el desarrollo de este proyecto se eligió **JavaScript** debido a varias razones clave que lo hacen una opción ideal para sistemas web.

- **Popularidad y Comunidad:** JavaScript es uno de los lenguajes de programación más populares en el desarrollo web. Esto significa que hay una gran comunidad de

desarrolladores, abundantes recursos educativos y una vasta cantidad de bibliotecas y frameworks disponibles, lo que facilita la resolución de problemas y la implementación de nuevas funcionalidades.

- **Interactividad y Dinamismo:** JavaScript permite crear interfaces de usuario interactivas y dinámicas. Con frameworks como React, Angular y Vue.js, es posible desarrollar aplicaciones web complejas y de alto rendimiento que ofrecen una experiencia de usuario fluida y atractiva.
- **Ecosistema de Herramientas:** El ecosistema de JavaScript incluye herramientas modernas de desarrollo como Webpack, Babel y ESLint, que mejoran la productividad y la calidad del código. Además, la integración con APIs y servicios externos es sencilla y robusta.
- **Escalabilidad:** JavaScript, junto con sus frameworks y bibliotecas, permite desarrollar aplicaciones escalables que pueden crecer y adaptarse a las necesidades cambiantes del negocio. Esto es crucial para un proyecto que busca reemplazar sistemas limitados y mejorar la eficiencia de la gestión de información.
- **Compatibilidad y Soporte:** JavaScript es compatible con todos los navegadores modernos y tiene un excelente soporte para tecnologías web emergentes. Esto asegura que las aplicaciones desarrolladas sean accesibles y funcionales en una amplia variedad de dispositivos y plataformas.

I.4.2 Framework de desarrollo

Un framework es un conjunto de reglas y convenciones que se usan para desarrollar software de manera más eficiente y rápida. Estos marcos de trabajo se emplean para ahorrar tiempo y esfuerzo en el desarrollo de aplicaciones, ya que proporcionan una estructura básica que se puede utilizar como punto de partida. Además, los frameworks también ofrecen soluciones a problemas comunes en el desarrollo de software, lo que significa que los desarrolladores pueden centrarse en las funcionalidades específicas de su aplicación en lugar de perder tiempo resolviendo problemas técnicos.(Lucena, 2023).

Frameworks más populares y utilizados en la actualidad

- **React:** Desarrollado por Facebook, es un framework de JavaScript muy popular para la creación de interfaces de usuario interactivas y dinámicas. Su enfoque en componentes

reutilizables y su eficiencia en la actualización del DOM lo hacen ideal para aplicaciones complejas.(Cyberstream, 2024)

- **Angular:** Respaldado por Google, Angular es un framework robusto para el desarrollo de aplicaciones web de una sola página (SPA). Ofrece una arquitectura basada en componentes y una amplia gama de herramientas integradas para el desarrollo y pruebas(Cyberstream, 2024).
- **Vue.js:** Conocido por su simplicidad y flexibilidad, Vue.js es un framework progresivo que permite a los desarrolladores adoptar gradualmente sus características. Es ideal para proyectos tanto pequeños como grandes(Cyberstream, 2024).

Tabla 2. Frameworks de desarrollo web más utilizados

Característica	Vue.js	React	Angular
Desarrollador	Evan You	Facebook	Google
Lanzamiento	2014	2013	2010
Arquitectura	MVVM	Biblioteca basada en componentes	MVC
Lenguaje	JavaScript	JavaScript (JSX)	TypeScript
Curva de aprendizaje	Suave	Moderada	Pronunciada
DOM	Virtual DOM	Virtual DOM	Real DOM
Rendimiento	Alto	Alto	Medio
Tamaño	Ligero	Moderado	Pesado
Comunidad	Creciente	Muy grande	Grande
Ecosistema	Moderado	Muy flexible	Completo
Uso en empresas	Alibaba, Xiaomi	Facebook, Instagram, Airbnb	Microsoft, Google, IBM
Documentación	Excelente	Excelente	Buena

Elección del framework de desarrollo

Para el desarrollo del nuevo sistema de gestión de información de seguridad y protección de la empresa ESPAC se eligió a **React** como framework de desarrollo debido a varias

características que lo hacen destacar entre otros frameworks y bibliotecas de desarrollo web:

Comunidad

React cuenta con una de las comunidades más grandes y activas en el ámbito del desarrollo web. Esta comunidad no solo proporciona un vasto número de recursos, tutoriales y ejemplos, sino que también asegura un soporte continuo y la rápida resolución de problemas. La colaboración y el intercambio de conocimientos dentro de la comunidad de React son invaluableles.

Ecosistema

El ecosistema de React es extremadamente flexible y robusto. Al ser una biblioteca enfocada en la construcción de interfaces de usuario, permite la integración con una amplia variedad de herramientas y librerías adicionales. Por ejemplo, para la gestión del estado, podemos utilizar Redux o MobX, y para el enrutamiento, React Router es una opción popular. Esta flexibilidad nos permite adaptar React a las necesidades específicas de nuestro proyecto sin estar atados a un conjunto rígido de herramientas.

Curva de Aprendizaje

React tiene una curva de aprendizaje moderada, lo que significa que los desarrolladores pueden comenzar a ser productivos rápidamente. Su enfoque en componentes reutilizables y su sintaxis JSX, que combina JavaScript con HTML, facilitan la comprensión y el desarrollo de interfaces complejas.

Documentación

La documentación de React es una de las mejores en el ámbito del desarrollo web. Está bien estructurada, es detallada y cubre desde conceptos básicos hasta avanzados. Esto no solo facilita el aprendizaje, sino que también sirve como una referencia constante para los desarrolladores experimentados. La calidad de la documentación asegura que cualquier duda o problema pueda ser resuelto de manera eficiente.

Además, el equipo de desarrollo ya tiene un amplio conocimiento y experiencia en el uso de React, esto permite una implementación más rápida y eficiente, ya que no se requiere tiempo adicional para aprender un nuevo framework. Toda la infraestructura de desarrollo de la empresa está basada en React. Mantener la consistencia en el uso de tecnologías facilita la integración y el mantenimiento de las aplicaciones, además de permitir una mejor colaboración entre diferentes equipos. React es conocido por su alto rendimiento gracias a su Virtual DOM,

que optimiza las actualizaciones de la interfaz de usuario. Además, su arquitectura facilita la escalabilidad de las aplicaciones, permitiendo añadir nuevas funcionalidades sin comprometer el rendimiento.

I.4.3 Entorno de programación

Visual Studio Code, una popular plataforma de desarrollo de código abierto creada por Microsoft. Este potente editor de código es conocido por su interfaz de usuario amigable, su amplia gama de extensiones y su integración con herramientas como Git y Docker.(Vidal, 2023).

La elección de Visual Studio Code (VS Code) como entorno de programación para el desarrollo del presente proyecto se basa en varias razones que lo hacen una opción destacada:

- **Extensiones y Personalización:** Una de las mayores fortalezas de VS Code es su ecosistema de extensiones. Los desarrolladores pueden personalizar su entorno de trabajo con extensiones que añaden funcionalidades específicas, como linters, depuradores, y herramientas de integración continua. Esto permite adaptar el editor a las necesidades particulares del proyecto.
- **Integración con Git:** VS Code tiene una integración nativa con Git, lo que facilita el control de versiones y la colaboración en equipo. Los desarrolladores pueden realizar commits, push, pull y resolver conflictos directamente desde el editor.
- **Depuración y Herramientas de Desarrollo:** VS Code ofrece potentes herramientas de depuración que permiten a los desarrolladores identificar y solucionar errores de manera eficiente. Además, cuenta con terminal integrada, IntelliSense (autocompletado inteligente) y soporte para tareas automatizadas.
- **Rendimiento y Ligereza:** A pesar de ser un editor muy completo, VS Code es ligero y rápido. Esto es especialmente importante para mantener la productividad en proyectos grandes y complejos.
- **Comunidad y Documentación:** VS Code cuenta con una gran comunidad de usuarios y desarrolladores que contribuyen con extensiones, temas y soluciones a problemas comunes. Además, la documentación oficial es extensa y de alta calidad, lo que facilita el aprendizaje y la resolución de dudas.

- Gratuito y de Código Abierto: VS Code es gratuito y de código abierto, lo que lo hace accesible para cualquier desarrollador o equipo, independientemente de su presupuesto.

I.4.4 Metodología de desarrollo de software

Para el desarrollo del proyecto actual se decidió adoptar un enfoque ágil fundamentado en varias razones que alinean perfectamente con los objetivos y necesidades del proyecto:

- Adaptabilidad y Flexibilidad: Las metodologías ágiles permiten adaptarse rápidamente a los cambios en los requisitos y prioridades del proyecto. Dado que el desarrollo de un nuevo sistema web para ESPAC puede implicar ajustes frecuentes basados en feedback de los usuarios y cambios en las necesidades operativas, un enfoque ágil facilita la incorporación de estos cambios sin afectar significativamente el progreso del proyecto.
- Entrega Continua de Valor: Con un enfoque ágil, el desarrollo se realiza en iteraciones cortas, lo que permite entregar incrementos funcionales del software de manera regular. Esto asegura que los usuarios puedan empezar a beneficiarse de las nuevas funcionalidades lo antes posible, mientras se sigue mejorando y ampliando el sistema.
- Mejora Continua: Las metodologías ágiles promueven la reflexión y la mejora continua a través de reuniones regulares de revisión y retrospectiva. Esto permite al equipo identificar y solucionar problemas rápidamente, optimizando los procesos y mejorando la calidad del producto final.
- Colaboración y Comunicación: Un enfoque ágil fomenta la colaboración estrecha entre todos los miembros del equipo, incluyendo desarrolladores, responsables de negocio y usuarios finales. Esta comunicación constante asegura que el desarrollo esté alineado con las expectativas y necesidades reales de los usuarios, aumentando la probabilidad de éxito del proyecto.
- Enfoque en el Usuario: Las metodologías ágiles ponen un fuerte énfasis en la satisfacción del cliente y en la entrega de valor real. Al involucrar a los usuarios finales en el proceso de desarrollo, se asegura que el sistema desarrollado realmente resuelva los problemas y mejore la eficiencia de la gestión de información en ESPAC.

- Reducción de Riesgos: Al trabajar en ciclos cortos y entregar incrementos funcionales regularmente, se pueden identificar y mitigar riesgos de manera temprana. Esto reduce la probabilidad de grandes fallos y permite realizar ajustes antes de que los problemas se conviertan en críticos.

Tabla 3. Metodologías de desarrollo de software más utilizadas

Característica	Scrum	Kanban	Extreme Programming (XP)	Lean
Enfoque	Iterativo e incremental	Flujo continuo	Mejora continua y calidad	Eliminación de desperdicios
Estructura	Sprints y roles definidos	Tablero visual	Prácticas técnicas rigurosas	Optimización del proceso
Roles	Scrum Master, Product Owner, Equipo de desarrollo	No hay roles específicos	Coach, Cliente, Equipo de desarrollo	No hay roles específicos
Planificación	Planificación de sprints	Gestión de flujo	Planificación continua	Planificación basada en valor
Reuniones	Reuniones diarias, revisiones de sprint, retrospectivas	Reuniones según necesidad	Reuniones diarias, revisiones, retrospectivas	Reuniones según necesidad
Flexibilidad	Moderada	Alta	Alta	Alta
Documentación	Moderada	Mínima	Moderada	Mínima
Curva de aprendizaje	Moderada	Suave	Pronunciada	Moderada
Uso en empresas	Amplio	Creciente	Menor	Creciente
Ejemplos de uso	Desarrollo de software, proyectos complejos	Mantenimiento y soporte continuo	Proyectos de alta calidad técnica	Optimización de procesos empresariales

Programación Extrema es una metodología de desarrollo de software que se adapta a los postulados del Manifiesto ágil priorizando a la adaptabilidad y no el seguimiento de un plan. Básicamente esta metodología se centra en la prueba y error para el desarrollo de un producto de software funcional, permitiendo la participación activa del cliente en todo el proceso como condición fundamental para el resultado exitoso del proceso, promoviendo el trabajo en equipo e impulsando el buen clima laboral.(Salazar et al., 2018).

Comparación con otras metodologías ágiles

- Scrum: Aunque es una metodología ágil poderosa y ampliamente utilizada, Scrum se centra más en la gestión del proyecto y la organización del equipo, siendo más genérico y aplicable a diversos tipos de proyectos (no solo de software).
- Kanban: Es excelente para la visualización del flujo de trabajo y la mejora continua, pero no proporciona las mismas prácticas técnicas específicas para asegurar la calidad del código como lo hace XP.
- Lean: Aunque también se enfoca en la eficiencia y la eliminación de desperdicios, su aplicación es más amplia y no necesariamente está centrada en las prácticas de desarrollo de software de manera tan específica como XP.

Fundamentación del uso de la metodología XP

Se decidió seguir los pasos que propone XP para el desarrollo del software ya que proporciona un marco estructurado y disciplinado que permite una entrega rápida y continua de software de alta calidad, adaptándose a los cambios y asegurando una comunicación efectiva con los usuarios finales. Estas características hacen que XP sea una elección óptima para el desarrollo del sistema web destinado a mejorar la gestión de información de seguridad y protección en ESPAC.

Fases de la metodología XP

- **Planificación:** va de acuerdo con las historias de usuario, estas se priorizan y descomponen en mini versiones, luego la planificación se va revisando cada dos semanas aproximadamente, después de las iteraciones, para obtener un software útil, funcional, listo para las pruebas y su lanzamiento.

- **Diseño:** se trabaja con un código sencillo, realizando lo mínimo necesario para que funcione, se obtiene el prototipo. Luego para el diseño del software si va orientado a objetos se generan tarjetas CRC (Clase-Responsabilidad-Colaboración).
- **Codificación:** se hace a dos manos, es decir en parejas frente al mismo ordenador, en algunos casos se intercambian las parejas, para asegurar que el código sea más universal, de forma que cualquier otro trabajador pueda trabajar en él y entenderlo. Debe parecer que fue hecho por una sola persona, para que se obtenga una programación organizada y planificada.
- **Pruebas:** deben ser automáticas y continuas, esto es clave para proyectos a corto plazo. Incluso el mismo cliente puede hacer pruebas, proponer pruebas nuevas y validar las mini versiones.
- **Lanzamiento:** si se llega a este punto es porque se han probado todas las historias de usuario o mini versiones y han tenido éxito, ajustándose a los requerimientos del cliente. Generando un software útil y que puede incorporarse en el producto.(Technologies, 2022)

Conclusiones del capítulo

A partir del análisis realizado en este documento, se han derivado las siguientes conclusiones:

- Se tomó la decisión de desarrollar un sistema web para la empresa ESPAC que gestione la información de seguridad y protección que reemplace la versión actual del sistema GISP y las hojas de cálculos de Excel.
- El estudio de sistemas homólogos permitió la obtención de características y oportunidades de mejora para un nuevo sistema de gestión de información de seguridad y protección.
- La elección de JavaScript como lenguaje de programación y React como framework de desarrollo, combinada con la adopción de la metodología XP, se ajusta perfectamente a las necesidades del proyecto y promete ser una solución excelente.

CAPÍTULO II: DISEÑO Y CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE INFORMACIÓN COMO SOLUCIÓN AL PROBLEMA PLANTEADO

En este capítulo se aborda el diseño e implementación del sistema GISPV4, concebido como la solución integral al problema de gestión de información en la Empresa de Seguridad y Protección de la Aviación Civil (ESPAC). Partiendo de la metodología Programación Extrema (XP), se detallan las características y los requisitos tanto funcionales como no funcionales del sistema, asegurando una entrega continua y de alta calidad.

La elección de XP como metodología guía permite un enfoque iterativo y colaborativo, donde la retroalimentación constante y la adaptación a los cambios son fundamentales. Este enfoque garantiza que el desarrollo de GISPV4 se mantenga alineado con las necesidades reales de los usuarios y las exigencias operativas de ESPAC.

A lo largo de este capítulo, se describen los patrones de diseño adoptados y la arquitectura del sistema, proporcionando una visión clara y estructurada de cómo GISPV4 se construye para ser robusto, escalable y eficiente. Se exploran las decisiones de diseño clave y cómo estas contribuyen a la solución del problema planteado, asegurando que el sistema no solo cumpla con los requisitos actuales, sino que también esté preparado para futuras expansiones y mejoras.

II.1 Presentación de la solución

Para resolver el problema planteado, el nuevo sistema debe incorporar una serie de funcionalidades administrativas esenciales. En primer lugar, debe permitir el registro detallado de los datos de las unidades de la empresa, incluyendo la identificación y descripción de cada unidad, así como las posiciones específicas dentro de cada una. Además, es crucial que el sistema registre información sobre los agentes de seguridad y protección, incluyendo sus datos personales, roles y el grupo de trabajo al que pertenecen.

Una característica destacada del sistema será la inclusión de un mapa interactivo de Cuba. Este mapa permitirá a los operadores interactuar directamente con las posiciones de cada unidad, facilitando la asignación de agentes de seguridad a ubicaciones específicas, así como el registro de incidencias. Además, el sistema debe ofrecer herramientas de administración

avanzadas que permitan la actualización y gestión continua de los datos, asegurando que la información esté siempre actualizada y sea fácilmente accesible para los usuarios autorizados. Para la comunicación con los datos, el sistema se conectará a una API REST utilizando la biblioteca Axios.

El sistema debe estandarizar toda la información posible a través de nomencladores. Esto implica que todos los datos relevantes y repetitivos dentro de la aplicación deben ser gestionados mediante conjuntos de datos predefinidos y estructurados, conocidos como nomencladores.

Mapa Interactivo de Cuba

Desde la página de inicio, se debe mostrar un mapa interactivo del país de Cuba con las posiciones identificadas. Desde el propio mapa, los usuarios deben poder registrar datos de incidencias, medios y asignación de agentes en las posiciones correspondientes.



GISPv4

Incidencias

Registro de incidencias



Fuerzas

Asignaciones de agentes a posición

Medios

Actualización de medios disponibles



Posiciones

Posiciones de la UEB



Figura 1. Solución. Mapa

Nomencladores: Registro de datos de administración.

La sección de nomencladores está diseñada para gestionar y administrar los datos fundamentales que se utilizan en la aplicación para el registro de información. Esta sección permite la creación, modificación y eliminación de nomencladores, que son conjuntos de datos estandarizados y estructurados que facilitan la consistencia y precisión en el registro de información.

- Estandarización de Datos: Los nomencladores aseguran que los datos ingresados sigan un formato y estructura consistentes, lo cual es esencial para mantener la integridad y calidad de la información.

- **Facilidad de Actualización:** Al centralizar la administración de datos, los nomencladores permiten actualizaciones rápidas y eficientes. Cualquier cambio en los datos maestros se refleja automáticamente en toda la aplicación.
- **Reducción de Errores:** Al utilizar nomencladores predefinidos, se minimizan los errores de entrada de datos, ya que los usuarios seleccionan opciones de listas desplegables en lugar de ingresar información manualmente.
- **Seguridad y Control:** Los nomencladores permiten definir permisos y roles específicos para la administración de datos, asegurando que solo usuarios autorizados puedan realizar modificaciones.



Figura 2. Propuesta de solución

- **Mapa:** Desde este apartado, el usuario debe seleccionar una unidad específica. Eventualmente, se debe permitir al usuario seleccionar una fecha diferente a la actual y

el turno de trabajo (diurno o nocturno). El sistema debe mostrar un mapa de Cuba centralizado en las coordenadas de la unidad seleccionada. Deben estar presentes en el mapa marcadores para cada posición relacionada con la unidad. Además, cada marcador de posición debe tener la opción de actualizar las asignaciones de agentes, incluyendo la cantidad de agentes asignados a dicha posición, registrar una nueva incidencia ocurrida en la posición seleccionada y actualizar los medios disponibles en dicha posición.

- **Incidencias:** Desde este apartado, se pueden buscar y acceder a toda la información de cada incidencia registrada en el sistema. Los usuarios tienen la opción de filtrar los resultados por unidad y rangos de fechas, lo que permite una búsqueda específica y eficiente. Además de visualizar la información detallada de cada incidencia, este apartado ofrece la funcionalidad de modificar los datos registrados, permitiendo actualizaciones en caso de que se necesiten correcciones o cambios en los detalles del incidente. También se incluye la opción de eliminar incidencias que hayan sido registradas incorrectamente o que ya no sean relevantes.
- **Reportes:** En este apartado, los usuarios pueden visualizar resúmenes detallados de las incidencias registradas en el sistema. Los reportes se pueden generar y organizar en base a diferentes criterios como las unidades específicas, los tipos de incidencias y las instalaciones de las unidades.
- **Nomencladores:** En este apartado, el sistema debe permitir el registro de datos que son usados de manera regular en la aplicación para registrar información. Esto incluye unidades, instalaciones, tipos de medios, tipos de incidencias, hechos de incidencias, grupos de trabajo y agentes de seguridad y protección. Además, el sistema debe permitir el registro, actualización y eliminación de estos datos.
- **Seguridad:** En el apartado de seguridad, se debe efectuar el registro de usuarios que tendrán acceso a la aplicación, así como definir nuevos roles, donde cada rol puede tener un grupo de permisos.

II.2 Historias de usuarios

Las historias de usuario se usan, en el contexto de la ingeniería de requisitos ágil, una herramienta de comunicación que combina las fortalezas de ambos medios: escrito y verbal.

Describen, en una o dos frases, una funcionalidad desde el punto de vista del usuario, con el lenguaje que éste emplearía. El foco está puesto en qué necesidades o problemas soluciona lo que se va a construir.(Menzinski et al., 2018)

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas(Letelier & Letelier, 2006a).

Durante el diseño del software, se decidió agrupar las historias de usuario por funcionalidades clave, tales como insertar, modificar y eliminar datos, en una única historia de usuario. Esta decisión se tomó con el objetivo de mejorar la cohesión y el contexto de cada historia de usuario, permitiendo al desarrollador comprender mejor el alcance completo de cada funcionalidad.

Tabla 4. Historia de usuario #2. Gestionar unidades

Historia de Usuario	
Numero: 2	Nombre: Gestionar unidades
Usuario: Director General	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Permitirá al director general de la empresa ESPAC añadir una nueva unidad al sistema introduciendo los siguientes datos: • Nombre de unidad • Latitud • Longitud Al mismo tiempo mostrará un listado con las unidades registradas y permitirá modificar o eliminar una unidad.	
Observaciones: Debe estar autenticado en el sistema para realizar las operaciones	

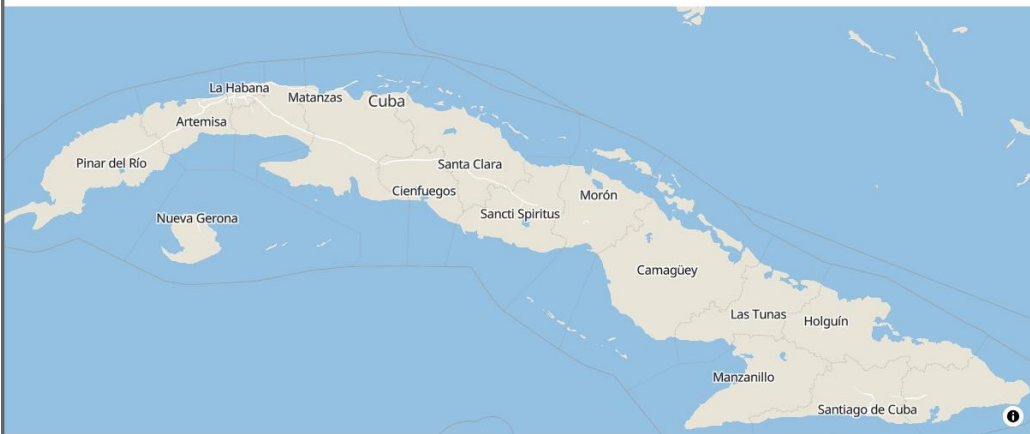
Prototipo de interfaz:

Añadir

> Varadero Matanzas		
> Camaguey		
> UEB: Jose Marti		

Añadir Unidad

Nombre Unidad *	Latitud *	Longitud *
-----------------	-----------	------------



CANCELAR **AÑADIR**

Tabla 5. Historia de usuario #3 Gestionar Instalaciones

Historia de Usuario	
Numero: 3	Nombre: Gestionar instalaciones
Usuario: Director General, director de unidad UEB	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Zamir Alarcón Cabrera	

Descripción: Permitirá al director general y director de unidad añadir una nueva terminal dado un identificador de unidad al sistema introduciendo los siguientes datos:

- Nombre de terminal

Al mismo tiempo mostrará un listado con las terminales registradas por cada unidad y permitirá modificar o eliminar una terminal.

Observaciones: Debe estar autenticado en el sistema para realizar las operaciones, debe existir alguna unidad registrada en el sistema.

Prototipo de interfaz:

Prototipo de interfaz para la gestión de terminales de una unidad. La interfaz muestra un encabezado con un menú desplegable que indica 'UEB: Jose Marti' y dos iconos de edición y eliminación. A continuación, se listan cinco terminales: 'Terminal 1', 'Terminal 2', 'Terminal 3', 'Aero Varadero T4' y 'Terminal 5'. Cada entrada de terminal incluye iconos de edición y eliminación. En la parte inferior del listado, hay un botón '+ Añadir'.

UEB: Jose Marti

Ingrese el nombre de Area/Termial que desea añadir a la unidad seleccionada

Terminal *

CANCELAR

AÑADIR

Tabla 6. Historia de usuario #12 Adicionar incidencia

Historia de Usuario	
Numero: 12	Nombre: Adicionar incidencia
Usuario: técnico de seguridad y protección, oficial de guardia de unidad UEB.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1.5	Iteración asignada: 2
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Permitirá al técnico de seguridad y protección y al oficial de guardia de la unidad UEB adicionar una incidencia a una posición dada, se mostrarán 3 formularios para completar el registro de la incidencia. Formulario 1 (Datos generales): • Unidad • Instalación • Posición • Tipo de incidencia • Hecho • Fecha • Hora Formulario 2 (Infractores): • Identificación • Nombres • Primer apellido • Segundo apellido • Dependencia • Opcion (eliminar infractor de la incidencia) • Botón (adicionar infractor) Formulario 3 (Datos del hecho): • Descripción • Observaciones • Agente	
Observaciones: Debe estar autenticado en el sistema para realizar las operaciones, cada incidencia puede contener uno o varios infractores.	
Prototipo de interfaz:	

Incidencia

1 DATOS GENERALES

2 INFRACTORES

3 DATOS DEL HECHO

Unidad *

UEB: Jose Marti

Instalacion *

Terminal 1

Posicion *

Aceso Vehicular entrada

Tipo de incidencia *

Relevante

Hecho *

Hecho relevante

Fecha

14/08/2024

Hora

11:15

← ATRAS

→ SIGUIENTE

Incidencia

✓ DATOS GENERALES

2 INFRACTORES

3 DATOS DEL HECHO

Identificacion

Nombres

Primer Apellido

Segundo Apel...

Dependencia

Identificacion

Nombres

Primer Apellido

Segundo Apel...

Dependencia

Identificacion

Nombres

Primer Apellido

Segundo Apel...

Dependencia

+ AÑADIR INFRACITOR

← ATRAS

→ SIGUIENTE

Incidencia

✓ DATOS GENERALES

✓ INFRACTORES

3 DATOS DEL HECHO

Descripcion

Observaciones

DATOS DEL AGENTE

Identificacion

A102478

Nombres

Juan

Primer Apellido

Perez

Segundo Apellido

Perez

← ATRAS

GUARDAR

II.3 Requisitos no funcionales

La calidad del software es un elemento fundamental para garantizar el éxito de un proyecto software, así como un factor competitivo esencial en las empresas que lo desarrollan.

Para mejorar y asegurar la calidad, algunas aproximaciones argumentan que es necesario tratar dicha calidad explícitamente desde el principio mediante su formalización en requisitos. Los requisitos de calidad (también llamados requisitos no-funcionales) son características, condiciones o restricciones con los que los ingenieros del software garantizan el correcto funcionamiento de un aplicativo. Los requisitos de calidad incluyen aspectos muy diversos, tales como disponibilidad, seguridad, rendimiento, escalabilidad, portabilidad, etc. Los nuevos métodos de desarrollo ágil de software (p.e. XP, Scrum, Kanban) abren paso a nuevos retos para poder garantizar un nivel de calidad satisfactorio de dichos aspectos. (Salamea Bravo et al., 2020).

RnF1-Software

RnF1.1- El Sistema deberá de funcionar sobre cualquier distribución de sistema operativo Linux y/o Windows 10 o superior.

RnF2-Hardware

RnF2.1-El sistema deberá requerir de 2 GB de memoria RAM.

RnF3-Interfáz o apariencia externa

RnF-3.1-La aplicación debe tener un diseño limpio y organizado, menús claros y accesibles, y una navegación consistente en todas las secciones de la aplicación."

RnF3.2-Todos los campos de entrada en los formularios deben tener etiquetas claras que indiquen la información requerida.

RnF3.3-La aplicación debe utilizar una paleta de colores consistente que incluya azul, blanco, rojo y amarillo.

RnF4-Rendimiento

RnF4.1-La aplicación debe cargar y responder a las acciones del usuario en menos de 240 milisegundos bajo condiciones normales de uso.

II.4 Planificación de las entregas

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la "velocidad" de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. (Letelier & Letelier, 2006b)

Estimación de esfuerzo por historia de usuario

Tabla 7. Estimación de esfuerzo por historia de usuario

Historias de usuarios		Puntos estimados
1	Iniciar sesión	0.5
2	Gestionar Unidades	0.5
3	Gestionar Instalaciones	0.5
4	Gestionar Posiciones	0.8
5	Gestionar Grupos	0.4
6	Gestionar Agentes	0.8
7	Gestionar Medios	0.3
8	Gestionar Configuración de Medios	0.6
9	Listar Posiciones en el mapa	1
10	Actualizar Fuerzas de Posición	1
11	Gestionar Medios Disponibles en posición	1
12	Adicionar Incidencia	1.5
13	Listar incidencias	1
14	Generar gráfico de reporte de incidencias	1

Plan de Iteraciones

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación. Asimismo, para cada historia de usuario se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores.(Collazo & Díaz, 2013).

Tabla 8. Plan de iteraciones

Iteración	Historias de Usuarios		Duración en semanas
1	1	Iniciar sesión.	2.7
	2	Gestionar unidades	
	3	Gestionar Instalaciones.	
	4	Gestionar posiciones.	
	5	Gestionar grupos.	
2	1	Gestionar agentes.	2.7
	2	Gestionar medios.	
	3	Gestionar configuración de medios.	
	4	Listar posiciones en el mapa.	
3	1	Actualizar fuerzas de posición.	5.5
	2	Adicionar incidencias.	
	3	Listar incidencias.	
	4	Gestionar medios disponibles en posición.	
	5	Generar gráfico de reporte de incidencias.	
Total			10.9

Plan de entregas

Tabla 9. Plan de entregas

Entregable	Iteración 1	Iteración 2	Iteración 3
Fecha de inicio	01/07/2024	22/07/2024	12/08/2024
Fecha de entrega	19/07/2024	09/08/2024	01/10/2024

II.5 Diseño del Sistema

Con la llegada del desarrollo de software, los profesionales en este campo han buscado una solución para disminuir la complejidad presente en el mantenimiento y organización del código. A través de la experiencia y el análisis de distintos sistemas informáticos se ha observado una serie repetitiva de elementos que son necesarios para el desarrollo de un sistema, al abstraer estos elementos, se da origen a los patrones de diseño. Un patrón de diseño es una técnica para resolver problemas simples y comunes que se encuentran en la vida diaria del desarrollo de software, sobre todo constituye una solución respecto al desarrollo de interacciones o interfaces.(Gavilánez Álvarez et al., 2022)

La Arquitectura de Software (AS), en tanto, describe la solución de un sistema y por lo tanto tradicionalmente se piensa como una parte temprana de la fase de diseño ya que reúne todos los requerimientos técnicos y operacionales y que son difíciles de cambiar durante el proceso de desarrollo. Es principalmente importante para satisfacer los requisitos no funcionales, que están relacionados a los atributos de calidad como el rendimiento, seguridad y escalabilidad.(Navarro et al., 2018).

II.5.1 Patrón de arquitectura

El desarrollo de software es un proceso no trivial que debe estar soportado en una estructura arquitectónica adecuada. Surgen así diferentes propuestas arquitectónicas como el modelo-cliente-servidor centralizado o distribuido, la arquitectura de n-capas, la arquitectura por componentes y la Arquitectura Orientada a Servicios - SOA, entre otras(Lemus, 2015).

Flux

Flux está diseñado específicamente para React. Es una arquitectura de aplicación destinada a evitar el concepto de flujo de datos y enlace multidireccional, que es común en los marcos

de trabajo MVC típicos. En su lugar, ofrece un flujo de datos unidireccional donde React es la capa de interfaz de usuario en el medio (Gackenhaimer, 2015).

El flujo de datos unidireccional facilita el seguimiento de los cambios en la aplicación y los hace más predecibles. Las preocupaciones están separadas: los componentes de la arquitectura Flux tienen roles claros y están fuertemente desacoplados. Funciona bien con la programación declarativa ya que el almacén puede actualizar la vista sin definir cómo cambiar estados (Chen et al., 2019).

Componentes de Flux:

Actions (Acciones): Son objetos simples que contienen información sobre cualquier interacción del usuario o evento que ocurra en la aplicación. Las acciones se envían a través de un despachador (dispatcher).

Dispatcher (Despachador): Es un mecanismo central que recibe todas las acciones y las envía a los almacenes correspondientes. Actúa como un centro de comunicación en la aplicación Flux.

Stores (Almacenes): Contienen el estado y la lógica de la aplicación. Los almacenes registran las acciones con el despachador y actualizan su estado en función de las acciones recibidas.

Views (Vistas): Son los componentes de React que renderizan la interfaz de usuario. Las vistas se suscriben a los cambios en los almacenes y se actualizan automáticamente cuando los datos en los almacenes cambian.

En el desarrollo de nuestra solución, decidimos seguir el patrón Flux para asegurar un flujo de datos unidireccional, lo cual mejora la trazabilidad y previsibilidad de los cambios en la aplicación. Para implementar este patrón, elegimos apoyarnos en la biblioteca Redux, que se ha consolidado como una herramienta poderosa y eficiente en la gestión del estado en aplicaciones React.

Representación arquitectónica

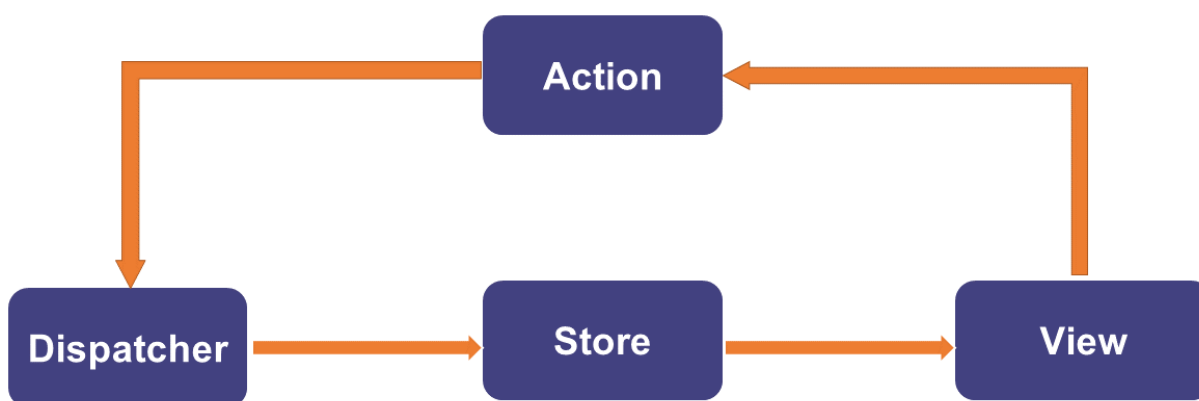


Figura 3. Representación arquitectónica

II.5.2 Patrones de diseño

En el desarrollo de aplicaciones de software, los patrones de diseño juegan un rol fundamental al proporcionar soluciones probadas y eficientes a problemas recurrentes en el diseño de sistemas. Estos patrones no solo facilitan la resolución de problemas específicos, sino que también contribuyen a la creación de código más limpio, mantenible y escalable.

Patrones GRASP

GRASP es el acrónimo de *General Responsibility Assignment Software Patterns* o, traducido al español, patrones de asignación de responsabilidad general. aunque muchos son más principios que patrones, nos indican reglas o técnicas para asignar responsabilidades en nuestro software con el fin de mantener bajo acoplamiento, conseguir alta cohesión y hacer nuestro software más escalable y mantenible.(Chamorro, 2024).

Experto: los componentes que manejan datos específicos, como el componente que gestiona la información de las unidades de la empresa y los agentes de seguridad, fueron diseñados

para ser los expertos en esa información. Esto asegura que cada componente tiene la responsabilidad de manejar los datos que conoce mejor, facilitando la lógica de negocio y el acceso a la información.

Componente de Gestión de Unidades:

Responsabilidad: Este componente es responsable de manejar toda la información relacionada con las unidades de la empresa. Esto incluye detalles como ubicación e instalaciones.

Funciones:

- Registrar Unidad: Función para agregar una nueva unidad al sistema con toda la información relevante.
- Actualizar Información de Unidad: Función para modificar los detalles de una unidad existente.
- Consultar Estado de Unidad: Función para recuperar el estado actual de una unidad específica.

Creador: Los componentes padres en la aplicación, como el componente que maneja el mapa interactivo de Cuba, son responsables de crear y gestionar los componentes hijos, como los marcadores de posición y las ventanas de formularios. Esto sigue el patrón de Creador, ya que el componente padre tiene la información necesaria para inicializar y configurar los componentes hijos.

Componentes Padres e Hijos

- Componente Padre: Mapa Interactivo:

Responsabilidad: Este componente es responsable de mostrar el mapa y gestionar los componentes hijos, como los marcadores de posición y las ventanas de formularios.

Funciones Principales:

Añadir Marcador: Función para crear y añadir un marcador en una ubicación específica del mapa.

Abrir Formulario: Función para abrir una ventana de formulario cuando se hace clic en un marcador.

- Componente Hijo: Marcador de Posición:

Responsabilidad: Representa una posición en el mapa.

Funciones Principales:

Mostrar una ubicación específica en el mapa.

Disparar eventos al ser clicado, como abrir un formulario.

- Componente Hijo: Ventana de Formulario:

Responsabilidad: Proporciona una interfaz para ingresar y mostrar información sobre una posición específica.

Funciones Principales:

Recoger y validar la información ingresada por el usuario.

Actualizar los datos del marcador asociado.

Bajo acoplamiento: Se ha minimizado el acoplamiento entre componentes utilizando props y contextos para pasar datos y funciones. Esto permite que los componentes sean más independientes y reutilizables, facilitando el mantenimiento y la escalabilidad de la aplicación.

Alta cohesión: Los componentes de la aplicación están diseñados para tener responsabilidades específicas y bien definidas. Por ejemplo, un componente puede estar dedicado exclusivamente a la visualización del mapa, mientras que otro maneja la lógica de autenticación. Esta alta cohesión asegura que cada componente se enfoque en una tarea particular, mejorando la claridad y la eficiencia del código.

Controlador: Los componentes de alto nivel, como el componente principal de la aplicación, actúan como controladores que manejan la navegación y la coordinación entre diferentes vistas y componentes. En este caso, se utiliza React Router DOM para manejar la navegación, lo que permite una gestión eficiente de las rutas y vistas dentro de la aplicación. Este patrón asegura que la lógica de control está centralizada, facilitando la gestión de eventos y la interacción entre componentes.

Estos patrones ayudaron a asignar responsabilidades de manera efectiva, asegurando que cada componente tenga un propósito claro y definido.

Conclusiones del capítulo

En este capítulo, se ha llevado a cabo un análisis exhaustivo y detallado de las historias de usuarios, los requisitos no funcionales y el diseño de la aplicación, siguiendo los principios de la metodología Extreme Programming (XP). A continuación, se presentan las conclusiones más relevantes:

- La definición de historias de usuarios ha permitido capturar de manera precisa las necesidades y expectativas de los usuarios finales. Este enfoque centrado en el usuario ha facilitado la priorización de funcionalidades y ha asegurado que el desarrollo se alinee con los objetivos del negocio.
- Se han identificado y documentado los requisitos no funcionales críticos para el éxito del proyecto, tales como la seguridad, la escalabilidad y el rendimiento. Estos requisitos aseguran que la aplicación no solo cumpla con las funcionalidades esperadas, sino que también ofrezca una experiencia de usuario robusta y confiable.
- El diseño de la aplicación se ha realizado siguiendo los principios de XP, lo que ha permitido una mayor flexibilidad y adaptabilidad a los cambios.
- La metodología XP ha demostrado ser una herramienta eficaz para gestionar el desarrollo del proyecto, promoviendo la comunicación constante entre el equipo de desarrollo y las partes interesadas.

CAPÍTULO III: IMPLEMENTACION Y VALIDACION DEL SISTEMA GISPV4

En este capítulo, se abordará la implementación del sistema y su posterior validación, siguiendo un enfoque estructurado y metódico basado en las prácticas de Extreme Programming (XP). La implementación del sistema es una fase crucial en el ciclo de vida del desarrollo de software, donde las especificaciones y diseños previamente definidos se transforman en un producto funcional. Este proceso incluye la codificación, integración y configuración de los diferentes componentes del sistema.

La validación del sistema, por otro lado, es esencial para asegurar que el software cumple con los requisitos establecidos y funciona correctamente en su entorno operativo. Para ello, se llevarán a cabo diversas tareas de ingeniería y pruebas de software, alineadas con los principios de XP, que garantizan la calidad y fiabilidad del producto final.

La metodología XP, con su énfasis en la colaboración, la retroalimentación continua y la entrega incremental, será el marco de referencia para todas las actividades descritas. Este enfoque no solo facilita la detección temprana de errores y la mejora continua, sino que también asegura que el sistema desarrollado sea robusto, eficiente y alineado con las necesidades del usuario.

Al final del capítulo, se presentarán las conclusiones y lecciones aprendidas durante el proceso de implementación y validación, proporcionando una visión integral del desarrollo del sistema y su preparación para el despliegue en un entorno real.

III.1 Codificación

Esta fase es el núcleo de XP. El desarrollo se realiza en ciclos cortos y repetitivos, donde se codifica, prueba y revisa el software continuamente. La retroalimentación del cliente es esencial para ajustar y mejorar el producto.

Tareas de ingeniería

En la primera iteración se definieron un total de 10 tareas de ingeniería desglosadas a partir de las historias correspondientes a los nomencladores de unidad, instalación y posición, además de la autenticación de usuarios.

Tabla 10. Tarea de ingeniería 1

Tarea	
Número de tarea: 1	Número de historia de usuario: 1
Nombre de la tarea: Implementar la funcionalidad para la autenticación de usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 1 de julio de 2024	Fecha de fin: 1 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permite mostrar un formulario de inicio de sesión en el sistema donde el usuario ingresa sus credenciales del dominio <code>espac.avianet.cu</code>	

Tabla 11. Tarea de ingeniería 2

Tarea	
Número de tarea: 2	Número de historia de usuario: 2
Nombre de la tarea: Implementar la funcionalidad de añadir unidades UEB al sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 2 de julio de 2024	Fecha de fin: 2 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permite registrar una nueva unidad UEB al sistema. Al hacer clic en un botón se muestra un modal con un formulario de tres campos de textos, nombre de unidad, latitud y longitud, además de un mapa para permitir la selección de latitud y longitud haciendo clic en el mapa.	

Tabla 12. Tarea de ingeniería 8

Tarea	
Número de tarea: 8	Número de historia de usuario: 4
Nombre de la tarea: Implementar la funcionalidad para listar las posiciones de una unidad e instalación seleccionada.	

Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 3 de julio de 2024	Fecha de fin: 3 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa una vista con dos selectores para seleccionar una unidad y una instalación y muestra una tabla con un listado de sus posiciones.	

Tabla 13. Tarea de ingeniería 9

Tarea	
Número de tarea: 9	Número de historia de usuario: 4
Nombre de la tarea: Implementar la funcionalidad para editar o añadir una posición al sistema.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 4 de julio de 2024	Fecha de fin: 5 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permite mostrar un formulario para añadir o editar alguna posición al sistema teniendo seleccionado previamente una unidad e instalación específica. Al seleccionar la opción para editar, el formulario debe cargar los datos de la posición seleccionada en los controles.	

Para una segunda iteración se definieron un total de 11 tareas de ingeniería desglosadas a partir de las historias de usuario correspondientes a los nomencladores de agentes, medios y grupos, además de la representación de posiciones en un mapa de Cuba.

Tabla 14. Tarea de ingeniería 20

Tarea	
Número de tarea: 20	Número de historia de usuario: 12
Nombre de la tarea: Implementar la funcionalidad de mostrar un mapa de Cuba centralizado en las coordenadas de una unidad.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de inicio: 22 de julio de 2024	Fecha de fin: 24 de julio de 2024

Programador responsable: Zamir Alarcón Cabrera
Descripción: Se implementa una vista que muestra un selector para seleccionar la unidad, un selector de fecha y un selector de turno, además de un mapa de Cuba centralizado en las coordenadas de la unidad seleccionada.

Tabla 15. tarea de ingeniería 21

Tarea	
Número de tarea: 21	Número de historia de usuario: 12
Nombre de la tarea: Implementar la funcionalidad de mostrar las posiciones en el mapa.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de inicio: 25 de julio de 2024	Fecha de fin: 29 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permite mostrar un marcador por cada posición de la unidad seleccionada en el mapa. Cada marcador debe reaccionar al evento clic mostrando un mensaje emergente con el nombre de la posición y las opciones para actualizar fuerzas, medios y añadir incidencia.	

Para una tercera iteración se definieron un total de 9 tareas de ingeniería desglosadas a partir de las historias de usuario correspondientes con la asignación de agentes a posiciones, control de medios disponibles por posición y registro de incidencias.

Tabla 16. Tarea de ingeniería 24

Tarea	
Número de tarea: 24	Número de historia de usuario: 13
Nombre de la tarea: Implementar la funcionalidad de actualizar las fuerzas de una posición en una fecha y turno dado.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de inicio: 21 de agosto de 2024	Fecha de fin: 23 de julio de 2024

Programador responsable: Zamir Alarcón Cabrera
Descripción: Se implementa esta funcionalidad que permite mostrar un formulario para actualizar la cantidad de agentes asignados a una posición dada para una fecha y turno dado.

Tabla 17. Tarea de ingeniería 26

Tarea	
Número de tarea: 26	Número de historia de usuario: 15
Nombre de la tarea: Implementar un contexto para el registro de incidencias	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 2 de septiembre de 2024	Fecha de fin: 4 de septiembre de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa un contexto que permitirá el acceso fácil a todos los nomencladores de la aplicación para utilizarlos en el formulario de registro de incidencias.	

III.2 Pruebas

En la metodología de Programación Extrema (XP), las pruebas de software son fundamentales y se integran en todo el proceso de desarrollo. Según XP se debe ser muy estricto con las pruebas. Solo se deberá liberar una nueva versión si esta ha pasado con el cien por ciento de la totalidad de las pruebas. En caso contrario se empleará el resultado de estas para identificar el error y solucionarlo con mecanismos ya definidos.(Echeverry Tobón & Delgado Carmona, 2007).

Alcance

- Componentes Probados: Se han realizado pruebas unitarias en los componentes principales del sistema, incluyendo formularios, botones, y componentes de visualización de datos.
- Funciones Probadas: Además de los componentes, se han probado funciones específicas como validaciones de formularios, cálculos y llamadas a APIs.

Herramientas Utilizadas:

- Framework de Pruebas: Jest
- Biblioteca de Pruebas: React Testing Library

Metodología:

- Configuración del Entorno de Pruebas: Se configuró Jest y React Testing Library en el entorno de desarrollo para ejecutar las pruebas.
- Escritura de Pruebas: Se escribieron pruebas unitarias para cada componente y función, asegurando que cada uno se comportara según lo esperado.
- Ejecución de Pruebas: Las pruebas se ejecutaron automáticamente en cada commit utilizando un sistema de integración continua (CI) para asegurar que no se introdujeran errores en el código base.

III.2.1 Pruebas unitarias

Estas pruebas se aplican a todos los métodos no triviales de todas las clases del proyecto con la condición que no se liberará ninguna clase que no tenga asociada su correspondiente paquete de pruebas. Uno de los elementos más importantes en estas es que idealmente deben ser construidas antes que los métodos mismos, permitiéndole al programador tener máxima claridad sobre lo que va a programar antes de hacerlo, así como conocer cada uno de los casos de pruebas que deberá pasar.(Echeverry Tobón & Delgado Carmona, 2007).

Las pruebas unitarias se ejecutan para comprobar el funcionamiento de los elementos de más bajo nivel de nuestro programa, puede ir desde probar el funcionamiento de una clase, hasta el funcionamiento de un método en particular.(Arias et al., 2018).

Las pruebas unitarias en XP son automatizadas, lo que permite ejecutarlas frecuentemente y detectar errores rápidamente. Esto es crucial para mantener la calidad del código y facilitar la integración continua. Durante el desarrollo del proyecto, se llevaron a cabo pruebas unitarias en tres iteraciones distintas, utilizando Jest y React Testing Library para automatizar el proceso.

```
4 describe('Gestion de unidades UEB', () => {
5   it('Debe devolver el estado inicial', () => {
6     expect(uebReducer(undefined, {})).toEqual({});
7   });
8
9   it('Listar unidades', () => {
10    const action = {
11      type: 'fetchUnidades',
12      payload: {
13        total: 10,
14        data: [{ id: 1, name: 'Unidad 1' }],
15        status: true,
16      },
17    };
18    const expectedState = {
19      total: 10,
20      data: [{ id: 1, name: 'Unidad 1' }],
21      errorMessage: null,
22      succes: true,
23    };
24    expect(uebReducer({}, action)).toEqual(expectedState);
25  });
26  it('adicionar unidad', () => {
27    const action = {
28      type: 'addUnidad',
29    };
30    const initialState = {
31      total: 10,
32      data: [{ id: 1, name: 'Unidad 1' }],
33      errorMessage: null,
34      succes: true,
35    };
36    expect(uebReducer(initialState, action)).toEqual(initialState);
37  });
38
39  it('Modificar unidad', () => {
40    const action = {
41      type: 'updateUnidad',
42    };
43    const initialState = {
44      total: 10,
45      data: [{ id: 1, name: 'Unidad 1' }],
46      errorMessage: null,
47      succes: true,
48    };
49    expect(uebReducer(initialState, action)).toEqual(initialState);
50  });
51 }
```

Figura 4. Pruebas unitarias al gestionar unidades UEB (Fuente: elaboración propia)

```
it('Adicionar instalacion', () => {
  const action = {
    type: 'addTerminalUnidad',
  };
  const initialState = {
    total: 10,
    data: [{ id: 1, name: 'Unidad 1' }],
    errorMessage: null,
    succes: true,
  };
  expect(uebReducer(initialState, action)).toEqual(initialState);
});

it('Eliminar instalacion', () => {
  const action = {
    type: 'deleteTerminal',
  };
  const initialState = {
    total: 10,
    data: [{ id: 1, name: 'Unidad 1' }],
    errorMessage: null,
    succes: true,
  };
  expect(uebReducer(initialState, action)).toEqual(initialState);
});

});
```

Figura 5. Pruebas unitarias al gestionar instalaciones (Fuente: elaboración propia)

```

4 describe('positionsReducer', () => {
5   it('Debe devolver el estado inicial', () => {
6     expect(positionsReducer(undefined, {})).toEqual({});
7   });
8
9   it('Listar todas las posiciones', () => {
10    const action = {
11      type: 'fetch_all_positions',
12      payload: {
13        data: [{ id: 1, name: 'Posicion 1' }],
14        count: 1,
15      },
16    };
17    const expectedState = {
18      data: [{ id: 1, name: 'Posicion 1' }],
19      count: 1,
20    };
21    expect(positionsReducer({}, action)).toEqual(expectedState);
22  });
23
24  it('Listar posiciones por unidad', () => {
25    const action = {
26      type: 'get_all_positions_by_unit',
27      payload: {
28        data: [{ id: 1, name: 'Posicion 1' }],
29      },
30    };
31    const expectedState = {
32      positions: [{ id: 1, name: 'Posicion 1' }],
33    };
34    expect(positionsReducer({}, action)).toEqual(expectedState);
35  });
36
37  it('Añadir posicion', () => {
38    const action = {
39      type: 'addPosition',
40    };
41    const initialState = {
42      data: [{ id: 1, name: 'Posicion 1' }],
43      count: 1,
44    };
45    expect(positionsReducer(initialState, action)).toEqual(initialState);
46  });
47

```

Figura 6. Pruebas unitarias al gestionar posiciones (Fuente: elaboración propia)

III.2.2 Resultados de las pruebas unitarias

Las pruebas unitarias se centraron en verificar la correcta renderización de los componentes y la configuración inicial adecuada. Se simuló la carga de datos para asegurar que los marcadores del mapa se renderizan correctamente. También se simuló la interacción con los marcadores del mapa para abrir un popup. Todas las pruebas se completaron con éxito, confirmando que los componentes individuales funcionan como se espera. Además, se probaron que las acciones de Redux se despacharon correctamente, manteniendo la integridad del estado global de la aplicación. A lo largo de tres iteraciones de pruebas, se evaluó la satisfacción y efectividad de las pruebas unitarias con los siguientes resultados:

Primera Iteración:

- Pruebas Completadas: 85%
- Satisfacción de Resultados: 80%
- Observaciones: Se identificaron algunos errores menores en la configuración inicial de los componentes, que fueron corregidos para la siguiente iteración.

Segunda Iteración:

- Pruebas Completadas: 90%
- Satisfacción de Resultados: 90%
- Observaciones: Mejoras significativas en la renderización y la interacción con los marcadores del mapa. Las acciones de Redux se despacharon de manera efectiva, asegurando la coherencia del estado.

Tercera Iteración:

- Pruebas Completadas: 100%
- Satisfacción de Resultados: 100%
- Observaciones: Todas las pruebas se completaron exitosamente sin errores. La integración y despachos de las acciones de Redux funcionaron a la perfección, asegurando la estabilidad y funcionalidad del sistema.

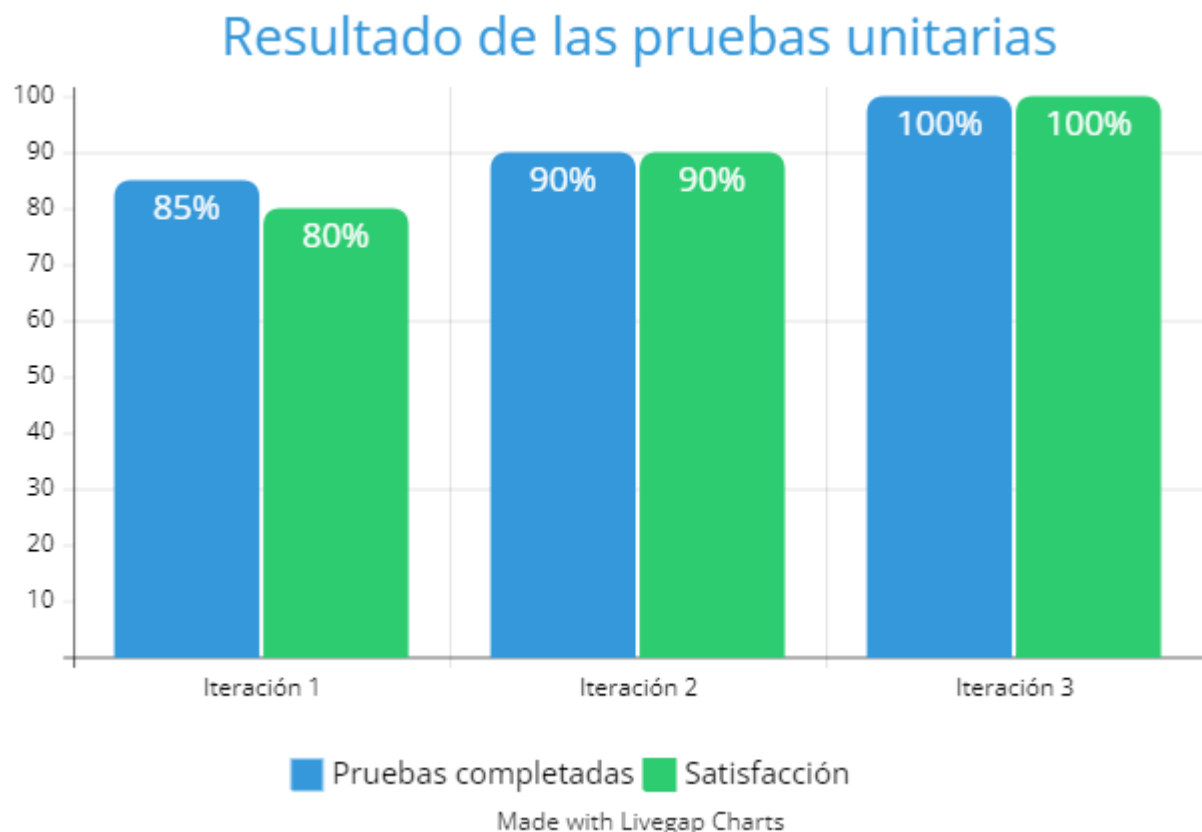


Figura 7. Resultado de las pruebas unitarias

III.2.3 Pruebas de aceptación

Las pruebas de aceptación son aquellas realizadas por los usuarios con carácter previo al paso a producción de una nueva versión del producto. Se trata de pruebas en un entorno de preproducción en la que se verifican si las funcionalidades pactadas para la entrega y recogidas en catálogos de requisitos, casos de uso, historias de usuario u otro hito documental, cumplen las expectativas del usuario. tienen como objetivo determinar si un sistema satisface o no sus criterios de aceptación y permitir al cliente decidir si aceptar o no el sistema. (Pardo Matos & Febles Estrada, 2014).

Para la primera iteración se definieron un total de 13 casos de pruebas de aceptación enfocadas en evaluar la gestión de unidades, de instalaciones y de posiciones, además de la autenticación de usuarios.

Caso de prueba de aceptación	
Código: HU2_P1	Historia de usuario: 2
Nombre: Adicionar unidad UEB	
Descripción: se prueba la funcionalidad para adicionar y/o modificar una unidad UEB en el sistema.	
Condiciones de ejecución: • El usuario debe estar previamente autenticado en el sistema. • El usuario debe tener acceso a los nomencladores del sistema.	
Pasos de ejecución: 1. El usuario entra en la vista unidades 2. Selecciona la opción Adicionar 3. Completa el formulario 4. Selecciona la opción Guardar	
Resultado: Satisfactorio	

Caso de prueba de aceptación	
Código: HU2_P2	Historia de usuario: 2
Nombre: Modificar unidad UEB	
Descripción: se prueba la funcionalidad para modificar una unidad UEB en el sistema.	
Condiciones de ejecución: • El usuario debe estar previamente autenticado en el sistema. • El usuario debe tener acceso a los nomencladores del sistema.	
Pasos de ejecución: 5. El usuario entra en la vista unidades 6. Selecciona la opción editar 7. Completa el formulario 8. Selecciona la opción Guardar	
Resultado: Satisfactorio	

Caso de prueba de aceptación	
Código: HU2_P3	Historia de usuario: 2
Nombre: Listar unidades UEB	

Descripción: se prueba la funcionalidad para listar las unidades UEB registradas en el sistema
Condiciones de ejecución: El usuario debe estar previamente autenticado en el sistema.
Pasos de ejecución: El usuario entra en la vista unidades
Resultado: Satisfactorio

Caso de prueba de aceptación	
Código: HU2_P4	Historia de usuario: 2
Nombre: Eliminar unidad UEB.	
Descripción: se prueba la funcionalidad para eliminar una unidad UEB registrada en el sistema.	
Condiciones de ejecución: - El usuario debe estar previamente autenticado en el sistema. - El usuario debe tener acceso a los nomencladores del sistema.	
Pasos de ejecución: - El usuario entra en la vista unidades - Selecciona una unidad del listado - Marca la opción eliminar - Marca la opción SI.	
Resultado: Satisfactorio	

Caso de prueba de aceptación	
Código: HU3_P1	Historia de usuario: 3
Nombre: Listar instalaciones	
Descripción: se prueba la funcionalidad para listar las instalaciones por unidades registradas en el sistema	
Condiciones de ejecución: - El usuario debe estar previamente autenticado en el sistema. - Debe existir al menos una unidad registrada en el sistema. - El usuario debe tener acceso a los nomencladores del sistema.	
Pasos de ejecución: - El usuario entra en la vista unidades - Despliega una unidad	
Resultado: Satisfactorio	

Para la segunda iteración se definieron un total de 14 casos de pruebas de aceptación enfocadas en evaluar la gestión de los medios, los agentes y las funcionalidades para mostrar las posiciones por unidades en un mapa de Cuba.

Caso de prueba de aceptación	
Código: HU6_P1	Historia de usuario: 6
Nombre: Listar grupos	
Descripción: se prueba la funcionalidad para listar los grupos registrados en el sistema	
Condiciones de ejecución: • El usuario debe estar previamente autenticado en el sistema.	
Pasos de ejecución: 1. El usuario entra en la vista grupos	
Resultado: Satisfactorio	

Caso de prueba de aceptación	
Código: HU6_P2	Historia de usuario: 6
Nombre: Adicionar grupos	
Descripción: se prueba la funcionalidad para adicionar grupos al sistema	
Condiciones de ejecución: • El usuario debe estar previamente autenticado en el sistema. • El usuario debe tener acceso a los nomencladores del sistema.	
Pasos de ejecución: 1. El usuario entra en la vista grupos 2. Marca la opción adicionar 3. Completa el formulario 4. Marca la opción Guardar	
Resultado: Satisfactorio	

Caso de prueba de aceptación	
Código: HU6_P3	Historia de usuario: 6

Nombre: Modificar grupos
Descripción: se prueba la funcionalidad para modificar un grupo registrado en el sistema
Condiciones de ejecución: • El usuario debe estar previamente autenticado en el sistema. • El usuario debe tener acceso a los nomencladores del sistema.
Pasos de ejecución: 1. El usuario entra en la vista grupos 2. Marca la opción modificar en un grupo 3. Completa el formulario 4. Marca la opción Guardar
Resultado: Satisfactorio

Para una tercera iteración se definieron un total de 6 casos de pruebas de aceptación enfocadas en evaluar la gestión de la configuración de medios por posición, la gestión de medios disponibles en cada posición, la gestión de las fuerzas en cada posición para una fecha y turno dado y el registro de incidencias.

Caso de prueba de aceptación	
Código: HU10_P1	Historia de usuario: 10
Nombre: Listar configuración de medios por posición	
Descripción: se prueba la funcionalidad para listar la configuración de los medios de cada posición	
Condiciones de ejecución: • El usuario debe estar previamente autenticado en el sistema. • Deben existir posiciones registradas en el sistema • Deben existir medios registrados en el sistema • El usuario debe tener acceso a la configuración del sistema.	
Pasos de ejecución: 5. El usuario entra en el menú configuración 6. El usuario entra en la vista medios	
Resultado: Satisfactorio	

Caso de prueba de aceptación	
Código: HU10_P2	Historia de usuario: 10

Nombre: Añadir un medio a la configuración de una posición
Descripción: se prueba la funcionalidad para añadir medios requeridos en una posición
Condiciones de ejecución: • El usuario debe estar previamente autenticado en el sistema. • Deben existir posiciones registradas en el sistema • Deben existir medios registrados en el sistema • El usuario debe tener acceso a la configuración del sistema.
Pasos de ejecución: 1. El usuario selecciona configuración en el menú 2. El Usuario entra en la vista medios 3. El Usuario selecciona una posición 4. Marca la opción adicionar medio 5. Completa el formulario 6. Selecciona Guardar
Resultado: Satisfactorio

III.3.4 Análisis de las pruebas de aceptación

Durante el desarrollo del software, se llevaron a cabo un total de 33 pruebas de aceptación distribuidas en 3 iteraciones, siguiendo la metodología de Programación Extrema (XP). Estas pruebas fueron fundamentales para asegurar que el sistema cumpliera con los requisitos y expectativas del cliente.

Cada iteración incluyó una serie de pruebas diseñadas para validar diferentes aspectos del sistema, desde la funcionalidad básica hasta características más avanzadas. Las pruebas de aceptación se planificaron y ejecutaron en colaboración con los usuarios finales, garantizando que cada funcionalidad desarrollada fuera revisada y aprobada antes de avanzar a la siguiente fase.



Figura 8. Resultados de las pruebas de aceptación. (Fuente: Elaboración propia)

Conclusiones del capítulo

El proceso de desarrollo y pruebas del nuevo sistema web para la gestión de información en ESPAC se llevó a cabo de manera estructurada y eficiente, siguiendo la metodología de Extreme Programming (XP). A continuación, se destacan las principales conclusiones:

- La utilización de historias de usuarios permitió una clara definición de los requisitos y expectativas del cliente. Este enfoque facilitó la priorización de tareas y la entrega de funcionalidades incrementales, asegurando que el sistema cumpliera con las necesidades reales de ESPAC.
- La utilización de Jest y React Testing Library para automatizar las pruebas unitarias fue crucial para alcanzar un alto nivel de calidad en el software. La iteración final no solo resolvió todos los problemas pendientes, sino que también mejoró significativamente la calidad del software, logrando así un 100% de satisfacción del cliente.
- Las pruebas de aceptación fueron fundamentales para validar que el sistema cumpliera con los requisitos funcionales y no funcionales establecidos.

CONCLUSIONES FINALES

El desarrollo del nuevo sistema web para la gestión de información en la Empresa de Seguridad y Protección de la Aviación Civil (ESPAC) fue un proceso integral que permitió abordar y solucionar las deficiencias del sistema anterior. A continuación, se presentan las conclusiones generales del proyecto:

- La transición de una aplicación web obsoleta y hojas de cálculo de Microsoft Excel a un sistema basado en React ha permitido una modernización tecnológica esencial. Esto ha mejorado la adaptabilidad del sistema a los nuevos requerimientos de información y operaciones diarias.
- La metodología XP ha sido crucial para el desarrollo del sistema, permitiendo una rápida adaptación a los cambios en los requisitos del cliente y mejorando la calidad del software entregado.
- La adopción de una interfaz de usuario intuitiva y funcionalidades mejoradas ha resultado en una mayor satisfacción de los usuarios finales. El nuevo sistema es más accesible y fácil de usar, lo que facilita la adopción y uso eficiente por parte del personal de la empresa.
- La implementación de prácticas como el desarrollo basado en historias de usuarios, las pruebas unitarias con Jest y las pruebas de aceptación, garantizó la entrega de un sistema robusto y confiable.

En resumen, el proyecto ha cumplido con sus objetivos al desarrollar un sistema web robusto y eficiente que mejora significativamente la gestión de la información en la Empresa de Seguridad y Protección de la Aviación Civil en Cuba. Este éxito se debe a la combinación de tecnologías modernas, metodologías ágiles y un enfoque centrado en el usuario.

RECOMENDACIONES

- Se recomienda la integración del nuevo sistema con el sistema eTES utilizado por la empresa ESPAC para la planificación y gestión empresarial creando un entorno de trabajo más cohesivo y eficiente.

REFERENCIAS BIBLIOGRÁFICAS

- Alfonso, B. M., Eusebio, M. O., Carlos, & Flavio, M. O., Juan. (2020). *Metodología de la investigación. Métodos y técnicas*. Grupo Editorial Patria.
- Andrada, L. R., & Santos, S. D. L. (2019). *Un análisis de los principales indicadores de calidad de los aeropuertos de España, Chile, Brasil y Perú*. ACCI (Asociación Cultural y Científica Iberoamericana).
- Arias, S. V., Soria, T. M., Moya, P. N., & Palma, P. M. (2018). Control de calidad del software mediante pruebas automatizadas de integración y pruebas unitarias. *Ciencia Digital*, 2(3), Article 3. <https://doi.org/10.33262/cienciadigital.v2i3.140>
- Author. (2023, abril 18). Agente de seguridad de aeropuerto: Requisitos, responsabilidades y características únicas. *Describiendo Profesiones*. <https://describiendoprof.com/agente-de-seguridad-de-aeropuerto-requisitos-responsabilidades-y-caracteristicas-unicas-de-la-profesion/>
- Camila, M. (2023). *La importancia de los sistemas de información en las empresas*. HubSpot. <https://blog.hubspot.es/marketing/sistemas-de-informacion-empresas>
- Chamorro, A. (2024). *Patrones GRASP (1): Experto de la información, creador, bajo acoplamiento y alta cohesión*. <https://albertochamorro.dev/blog/patrones-grasp-parte-1/>
- Chen, S., Thaduri, U., & Ballamudi, V. (2019). Front-End Development in React: An Overview. *Engineering International*, 7, 117-126. <https://doi.org/10.18034/ei.v7i2.662>
- Collazo, A., & Díaz, M. (2013). *La programación extrema*. Universidad De Las Ciencias Informáticas.
- CPD, E. (2023). *Vigilancia en aeropuertos*. CPD.
- Cyberstream. (2024, mayo 3). Los 10 frameworks más populares y utilizados en la actualidad: Guía completa. *Byron Vargas* ®. <https://www.byronvargas.com/web/cuales-son-los-frameworks-mas-utilizados/>

- Directivos, R. (2016, diciembre 23). Gestión de la información: Claves para entenderla | EAE. *El blog de retos para ser directivo | Desafíos de la Gestión Empresarial*. <https://retos-directivos.eae.es/gestion-de-la-informacion-claves-para-entenderla/>
- Echeverry Tobón, L. M., & Delgado Carmona, L. E. (2007). *Caso práctico de la metodología ágil XP al desarrollo de software*. <https://hdl.handle.net/11059/794>
- Gackenhaimer, C. (2015). *Introduction to React*. Apress.
- García, B. (2021). *Introducción a la gestión de la información y del conocimiento en la empresa*. 9.
- Gavilánez Álvarez, O. D., Layedra Larrea, N. P., & Ramos Valencia, M. V. (2022). Análisis comparativo de Patrones de Diseño de Software. *Polo del Conocimiento: Revista científico - profesional*, 7(7 (JULIO 2022)), 2146-2165.
- Lemus, A. C. N. (2015). ARQUITECTURA POR COMPONENTES JEE, UN CASO PRÁCTICO. *Revista GTI*, 14(38), Article 38.
- Letelier, P., & Letelier, P. (2006a, abril 15). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)* [Artículo]. www.cyta.com.ar/ta0502/v5n2a1.htm; *Técnica Administrativa* issn:1666-1680. http://www.cyta.com.ar/ta0502/b_v5n2a1.htm
- Letelier, P., & Letelier, P. (2006b, abril 15). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)* [Artículo]. www.cyta.com.ar/ta0502/v5n2a1.htm; *Técnica Administrativa* issn:1666-1680. http://www.cyta.com.ar/ta0502/b_v5n2a1.htm
- Lucena, P. (2023, mayo 6). ¿Qué es el framework? | 2024. *Maestrías y MBA*. <https://www.cesuma.mx/blog/que-es-el-framework.html>
- Menzinski, A., Sobrino, P., & Alvarez, R. (2018). *Historias de usuario*. V, 60.
- Navarro, M. E., Moreno, M. P., Aranda, J., Parra, L., & Rueda, J. R. (2018). *Arquitectura de software en el proceso de desarrollo ágil: Una perspectiva basada en requisitos significantes para la arquitectura*. XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste). <http://sedici.unlp.edu.ar/handle/10915/67795>

- Pardo Matos, J. M., & Febles Estrada, A. (2014). *Proceso de Pruebas de Aceptación de Software*.
<https://repositorio.uci.cu/jspui/handle/ident/8092>
- Pathak, A. (2022, junio 21). *Ciclo de vida del desarrollo de software (SDLC): Una guía completa* | Geekflare. Geekflare Spain. <https://geekflare.com/es/software-development-life-cycle-sdlc-guide/>
- Roberto. (2023, noviembre 15). La Gestión Aeroportuaria: Importancia y Funciones. *Viarium*.
<https://viariumgroup.com/blog/viarium-que-es-la-gestion-aeroportuaria/>
- Salamea Bravo, M. J., González Palacio, L., Oriol Hilari, M., & Farré Tost, C. (2020). *Estimación y priorización de requisitos no-funcionales para desarrollo de software: Estado del arte*. 150-157.
<https://upcommons.upc.edu/handle/2117/336760>
- Salazar, J. C., Casallas, Á. T., Linares, J. C., Lozano, A., & Valbuena, Y. L. (2018). Scrum versus XP: Similitudes y diferencias. *Tecnología Investigación y Academia*, 6(2), Article 2.
- Technologies, G. (2022, abril 13). *Cómo funciona la Metodología XP en el Desarrollo de Software*.
<https://ginzo.tech/como-funciona-metodologia-xp-desarrollo-software/>
- Vidal, S. (2023, noviembre 30). *¿Cuáles son las principales características de Visual Studio Code?*
▷→. <https://tecnobits.com/cuales-son-las-principales-caracteristicas-de-visual-studio-code/>
- Zendesk. (2023, septiembre 7). *Control de incidencias: 8 pasos para gestionar con éxito*. Zendesk.
<https://www.zendesk.com.mx/blog/gestion-incidencias-procedimiento/>

ANEXOS

Entrevista

Entrevistador: Zamir Alarcón Cabrera

Entrevistado: Técnico de seguridad y protección. Adrián Allegue Paredes

Fecha: 26/06/2024

Lugar: UEB José Martí

Preguntas:

Descripción General del Trabajo:

- ¿Podrías describir brevemente tus responsabilidades diarias como técnico de seguridad en el aeropuerto?

Requisitos de Información:

- ¿Qué tipo de información necesitas acceder regularmente para realizar tu trabajo de manera efectiva?
- ¿Cómo gestionas actualmente esta información?

Mapas y Posiciones:

- ¿Qué tan importante es para ti tener un mapa del aeropuerto con las posiciones de las instalaciones?
- ¿Qué características específicas te gustaría ver en este mapa (por ejemplo, zoom, capas, etiquetas)?

Control de Instalaciones:

- ¿Qué tipo de información necesitas sobre las instalaciones (por ejemplo, estado, ubicación, acceso)?
- ¿Cómo te gustaría que se presentara esta información en una aplicación?

Gestión de Agentes:

- ¿Cómo se asignan actualmente los agentes a las posiciones por turno de trabajo?
- ¿Qué información necesitas sobre los agentes (por ejemplo, nombre, turno, posición asignada)?

Tabla 18. Historia de usuario # 4. Gestionar posiciones

Historia de Usuario	
Numero: 4	Nombre: Gestionar posiciones
Usuario: Director General, director de unidad UEB, técnico de seguridad y protección	

Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 0.8	Iteración asignada: 1
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Permitirá al director general y director de unidad adicionar una posición dado un identificador de unidad y un identificador de instalación, a través de un formulario en forma de pasos mostrando el nombre de la unidad y la instalación seleccionada. El formulario permitirá introducir los siguientes datos: • Posición • Contratados • Conciliados • Latitud • Longitud Al mismo tiempo permitirá modificar o eliminar una posición, además de, mostrar un listado con las posiciones registradas en el sistema. Observaciones: Debe estar autenticado en el sistema para realizar las operaciones, debe existir alguna instalación registrada en el sistema. Prototipo de interfaz:	

Tabla 19. Historia de usuario #1. Iniciar sesión

Historia de Usuario	
Numero: 1	Nombre: Iniciar sesión
Usuario: Director General, director de unidad UEB, técnico de seguridad y protección, oficial de guardia de unidad UEB	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Zamir Alarcón Cabrera	

Descripción: Permitirá al director general y director de unidad UEB, técnico de seguridad y protección y al oficial de guardia de la unidad UEB autenticarse en el sistema introduciendo los siguientes datos:

- Nombre de usuario
- Contraseña

Observaciones: Las credenciales deben ser las utilizadas para acceder al dominio **espac.avianet.cu**

Prototipo de interfaz:

Tabla 20. Historia de usuario #5. Gestionar grupos

Historia de Usuario	
Numero: 5	Nombre: Gestionar grupos
Usuario: Director General, director de unidad UEB, técnico de seguridad y protección	
Prioridad en negocio: Alta	Riesgo en desarrollo: baja
Puntos estimados: 0.4	Iteración asignada: 1
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Permitirá al director general, director de unidad y al técnico de seguridad y protección de la unidad listar los grupos registrados en el sistema, además permitirá adicionar un grupo, así como modificar o eliminar un grupo.	
Observaciones: Debe estar autenticado en el sistema para realizar las operaciones	
Prototipo de interfaz:	

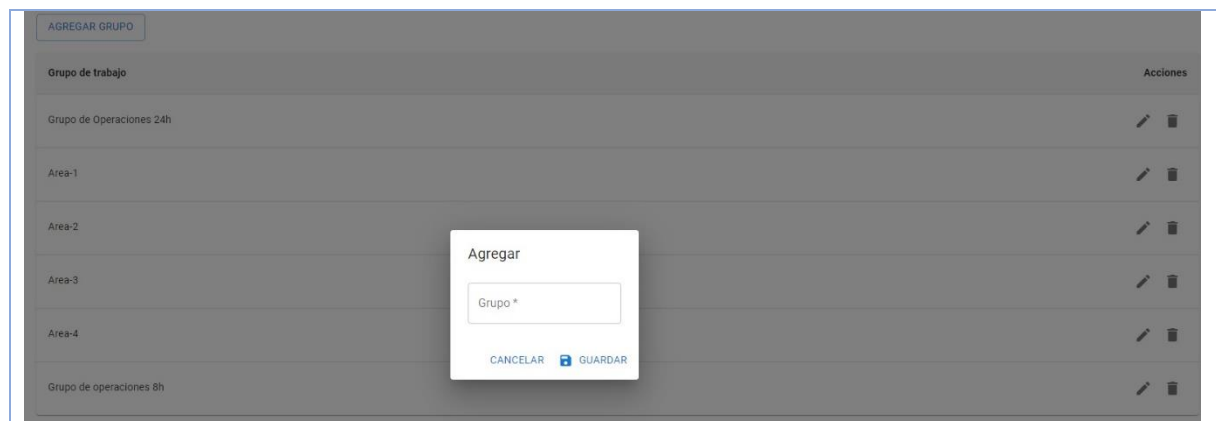


Tabla 21. Historia de usuario #6. Gestionar agentes

Historia de Usuario	
Numero: 6	Nombre: Gestionar agentes
Usuario: director de unidad UEB	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 0.8	Iteración asignada: 1
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Permitirá al director de unidad listar los agentes registrados para una unidad específica, además de, adicionar agentes al sistema dado un identificador de unidad, así como modificar un agente introduciendo los siguientes datos: • Nombres • Primer apellido • Segundo apellido • Numero de credencial • Grupo • Instalación	
Observaciones: Debe estar autenticado en el sistema para realizar las operaciones	
Prototipo de interfaz:	

Añadir ASP

Terminal

Grupo

Nombres *

Primer apellido *

Segundo apellido *

Identificación

Número de credencial

CANCELAR

GUARDAR Y CONTINUAR

GUARDAR

Tabla 22. Historia de usuario #7. Gestionar medios

Historia de Usuario	
Numero: 7	Nombre: Gestionar medios
Usuario: Director General, director de unidad UEB, técnico de seguridad y protección	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 0.3	Iteración asignada: 1
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Permitirá al director general, al director de UEB y al técnico de seguridad y protección listar los tipos de medios registrados en el sistema, además permite adicionar, modificar y eliminar medios registrados, mostrando los siguientes datos: - Medio - Descripción - Opciones (opciones de administración)	
Observaciones: Debe estar autenticado en el sistema para realizar las operaciones	
Prototipo de interfaz:	

+ AÑADIR MEDIO

Medio	Descripción	Opciones
Bastón		
Metor	Detector de metales	
Anteojos	Para la periferia	
Radio	Hytera	
Linterna		
Cinta	Cinta para sellar equipajes rotos o abiertos	
Libro de incidencias	RC-01785	

Añadir medio
×

Nombre del medio *

Descripción

CANCELAR
GUARDAR

Tabla 23. Historia de usuario #8. Gestionar configuración de medios

Historia de Usuario	
Numero: 8	Nombre: Gestionar configuración de medios
Usuario: Director General, director de unidad UEB, técnico de seguridad y protección	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 0.6	Iteración asignada: 1
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Permitirá al director general, al director de UEB y al técnico de seguridad y protección listar la configuración de los medios mostrando un listado de las posiciones de una unidad dada y un listado de la cantidad de cada medio necesaria para una posición seleccionada. De cada medio se muestra el nombre de la posición y la instalación a la que pertenece y de cada medio de una posición se muestra el nombre del medio y la cantidad requerida. También permite adicionar un medio a una posición y modificar la cantidad requerida.	
Observaciones: Debe estar autenticado en el sistema para realizar las operaciones	
Prototipo de interfaz:	

Unidad
UEB: Jose Marti

ANADIR MEDIO

Posicion	Entidad
☐ Acceso Vehicular entrada	Terminal 1
Pir de Trabajadores entrada	Terminal 1
P-45	Terminal 1
P-15	Terminal 1
Pir de Trabajadores P3	Terminal 1
P5-b	Terminal 1
P5-b	Terminal 1

Medio	Cantidad	Opciones
Motor	1	
Linterna	1	
Libro de Incidencias	2	
Bastón	1	

Configuracion de medio

Medio
Bastón

Cantidad
1

CANCELAR GUARDAR

Tabla 24. Historia de usuario #9. Listar posiciones en el mapa

Historia de Usuario	
Numero: 9	Nombre: Listar posiciones en el mapa
Usuario: director general, director de unidad UEB, técnico de seguridad y protección, oficial de guardia de unidad UEB.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Permitirá al director general, director de unidad UEB, técnico de seguridad y protección y al oficial de guardia de la unidad UEB mostrar en un mapa de Cuba la ubicación de cada posición de una unidad seleccionada. También mostrará un mensaje emergente en cada posición al hacer clic con las opciones para actualizar las fuerzas, los medios disponibles para una fecha y turno dados. Además, una opción para abrir un formulario para adicionar una incidencia en la posición seleccionada.	
Observaciones: Debe estar autenticado en el sistema para realizar las operaciones, el mapa debe estar centralizado en la unidad seleccionada.	
Prototipo de interfaz:	

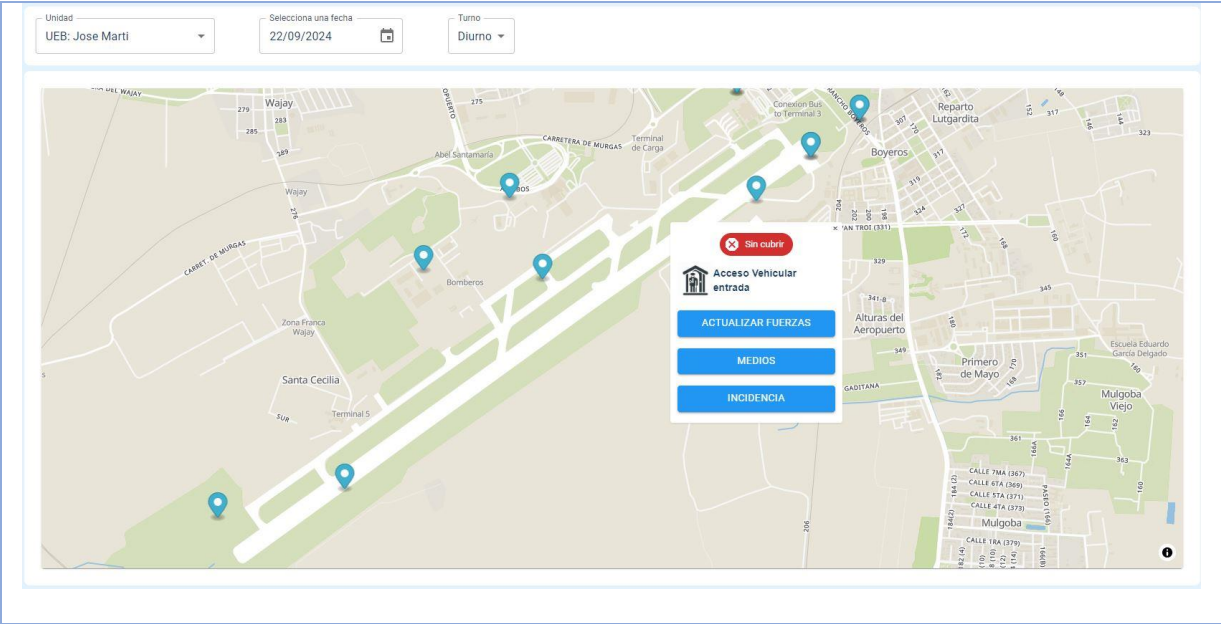


Tabla 25. Historia de usuario #10. Actualizar fuerzas de posición

Historia de Usuario	
Numero: 10	Nombre: Actualizar fuerzas de posición
Usuario: técnico de seguridad y protección, oficial de guardia de unidad UEB.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Permitirá al técnico de seguridad y protección y al oficial de guardia de la unidad UEB actualizar la cantidad de agentes asignados a la posición seleccionada, mostrando la cantidad de agentes contratados y conciliados de la posición.	
Observaciones: Debe estar autenticado en el sistema para realizar las operaciones	

Prototipo de interfaz:



Tabla 26. Historia de usuario #11. Gestionar medios disponibles

Historia de Usuario	
Numero: 11	Nombre: Gestionar medios disponibles
Usuario: técnico de seguridad y protección, oficial de guardia de unidad UEB.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Permitirá al técnico de seguridad y protección y al oficial de guardia de la unidad UEB listar los medios de una posición dada y actualizar la cantidad disponible de cada medio.	
Observaciones: Debe estar autenticado en el sistema para realizar las operaciones	

Prototipo de interfaz:

Medios en Acceso Vehicular entrada

Medio	Cantidad	Disponibles	Opciones
Metor	1	1	[Editar] [Eliminar]
Linterna	1	1	[Editar] [Eliminar]
Libro de incidencias	2	1	[Editar] [Eliminar]
Bastón	1	1	[Editar] [Eliminar]

Cancel OK

Configuración de medio

Medio: Linterna

Disponibles: 1

CANCELAR GUARDAR

(Detrás de la ventana de configuración se ve la parte inferior de la tabla de medios)

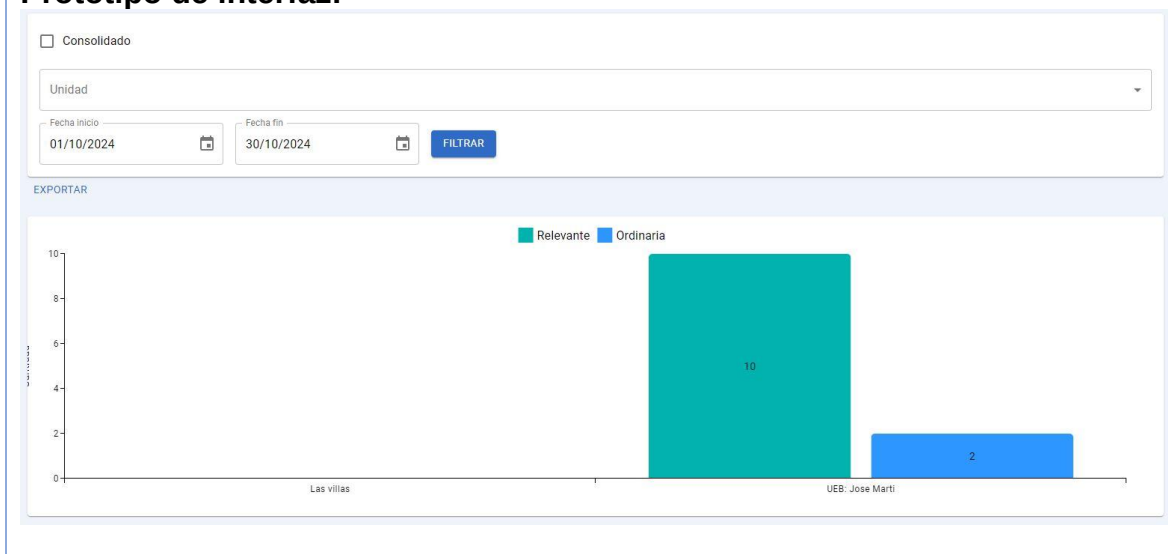
Linterna	1	1	[Editar] [Eliminar]
Libro de incidencias	2	1	[Editar] [Eliminar]
Bastón	1	1	[Editar] [Eliminar]

Tabla 27. Historia de usuario #13. Listar incidencias

Historia de Usuario	
Numero: 13	Nombre: Listar incidencias
Usuario: Director General, Técnico de seguridad de UEB, Oficial de guardia	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Permitirá al director general de la empresa, al técnico de seguridad de UEB y al oficial de guardia listar las incidencias registradas dada una unidad, una instalación y un rango de fecha.	
Observaciones: Debe estar autenticado en el sistema para realizar las operaciones	
Prototipo de interfaz:	
	

Tabla 28. Historia de usuario #14. Generar gráfico de reportes de incidencias

Historia de Usuario	
Numero: 14	Nombre: Generar gráfico de reporte de incidencia
Usuario: Director General, Técnico de seguridad de UEB, Oficial de guardia	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Permitirá al director general de la empresa, al técnico de seguridad de UEB y al oficial de guardia generar un gráfico que muestra la cantidad de incidencias de cada tipo de cada unidad, filtrado por rangos de fechas.	
Observaciones: Debe estar autenticado en el sistema para realizar las operaciones	

Prototipo de interfaz:**Tareas de ingeniería iteración 1**

Tarea	
Número de tarea: 3	Número de historia de usuario: 2
Nombre de la tarea: Implementar la funcionalidad de añadir y actualizar unidades UEB en el sistema.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 1 de julio de 2024	Fecha de fin: 1 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permite mostrar un formulario que contiene un mapa de Cuba para registrar o actualizar una unidad UEB al sistema, haciendo uso de la biblioteca MapLibreGL.	

Tarea	
Número de tarea: 4	Número de historia de usuario: 2
Nombre de la tarea: Implementar la funcionalidad que permite eliminar una unidad UEB del sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 1 de julio de 2024	Fecha de fin: 1 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	

Descripción: Se implementa esta funcionalidad que permite la eliminación de una unidad UEB registrada en el sistema, debe mostrar un mensaje de confirmación.

Tarea	
Número de tarea: 5	Número de historia de usuario: 3
Nombre de la tarea: Implementar la funcionalidad de listar las instalaciones por unidades.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 1 de julio de 2024	Fecha de fin: 1 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permite listar las instalaciones de cada unidad y muestra las opciones para editar, eliminar y añadir.	

Tarea	
Número de tarea: 6	Número de historia de usuario: 3
Nombre de la tarea: Implementar la funcionalidad añadir o actualizar instalaciones en el sistema.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 1 de julio de 2024	Fecha de fin: 1 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permite mostrar un formulario para añadir o editar alguna instalación.	

Tarea	
Número de tarea: 7	Número de historia de usuario: 3
Nombre de la tarea: Implementar la funcionalidad eliminar instalación del sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 1 de julio de 2024	Fecha de fin: 1 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permite eliminar una instalación registrada en el sistema, mostrando un mensaje de confirmación.	

Tarea	
Número de tarea: 10	Número de historia de usuario: 4
Nombre de la tarea: Implementar la funcionalidad para eliminar una posición del sistema.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 1 de julio de 2024	Fecha de fin: 1 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permite eliminar una posición registrada en el sistema, mostrando un mensaje de confirmación.	

Tareas de ingeniería iteración 2

Tarea	
Número de tarea: 11	Número de historia de usuario: 6
Nombre de la tarea: Implementar la funcionalidad de listar los grupos registrados en el sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 22 de julio de 2024	Fecha de fin: 22 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permite listar en una tabla los grupos registrados en el sistema, además muestra las opciones para adicionar, editar o eliminar.	

Tarea	
Número de tarea: 12	Número de historia de usuario: 6
Nombre de la tarea: Implementar la funcionalidad de añadir o actualizar grupos en el sistema.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 22 de julio de 2024	Fecha de fin: 23 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	

Descripción: Se implementa esta funcionalidad que muestra un campo de texto en un modal para añadir o modificar un grupo registrado en el sistema.

Tarea	
Número de tarea: 13	Número de historia de usuario: 6
Nombre de la tarea: Implementar la funcionalidad de eliminar un grupo registrado en el sistema.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 23 de julio de 2024	Fecha de fin: 23 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permite eliminar un grupo registrado en el sistema, mostrando un mensaje de confirmación.	

Tarea	
Número de tarea: 14	Número de historia de usuario: 8
Nombre de la tarea: Implementar la funcionalidad de listar los agentes registrados en el sistema.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de inicio: 24 de julio de 2024	Fecha de fin: 26 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa una vista que muestra un selector para seleccionar una unidad específica y una tabla con los agentes de esa unidad, además de las opciones para adicionar, actualizar y eliminar un agente.	

Tarea	
Número de tarea: 15	Número de historia de usuario: 7
Nombre de la tarea: Implementar la funcionalidad de añadir o actualizar agentes.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 1 de julio de 2024	Fecha de fin: 1 de julio de 2024
Programador responsable: Zamir Alarcón Cabrera	

Descripción: Se implementa esta funcionalidad que permite a través de un formulario añadir o modificar un agente al sistema teniendo previamente una unidad seleccionada.

Tarea

Número de tarea: 16

Número de historia de usuario: 7

Nombre de la tarea: Implementar la funcionalidad de eliminar un agente registrado en el sistema.

Tipo de tarea: Desarrollo

Puntos estimados: 0.3

Fecha de inicio: 28 de julio de 2024

Fecha de fin: 31 de julio de 2024

Programador responsable: Zamir Alarcón Cabrera

Descripción: Se implementa esta funcionalidad que permite eliminar un agente registrado en el sistema, mostrando un mensaje de confirmación.

Tarea

Número de tarea: 17

Número de historia de usuario: 9

Nombre de la tarea: Implementar la funcionalidad de listar los medios registrados en el sistema.

Tipo de tarea: Desarrollo

Puntos estimados: 0.2

Fecha de inicio: 31 de julio de 2024

Fecha de fin: 1 de agosto de 2024

Programador responsable: Zamir Alarcón Cabrera

Descripción: Se implementa una vista que muestra una tabla con los medios registrados en el sistema, además de las opciones para añadir, modificar o eliminar un medio.

Tarea

Número de tarea: 18

Número de historia de usuario: 9

Nombre de la tarea: Implementar la funcionalidad de añadir o actualizar un medio en el sistema.

Tipo de tarea: Desarrollo

Puntos estimados: 0.2

Fecha de inicio: 1 de agosto de 2024

Fecha de fin: 5 de agosto de 2024

Programador responsable: Zamir Alarcón Cabrera
Descripción: Se implementa una vista que muestra un modal con un formulario para añadir o modificar un medio.

Tarea	
Número de tarea: 19	Número de historia de usuario: 9
Nombre de la tarea: Implementar la funcionalidad de eliminar un medio registrado en el sistema.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 5 de agosto de 2024	Fecha de fin: 7 de agosto de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permite eliminar un medio registrado en el sistema, mostrando un mensaje de confirmación.	

Tareas de ingeniería iteración 3

Tarea	
Número de tarea: 22	Número de historia de usuario: 10
Nombre de la tarea: Implementar la funcionalidad de listar la configuración de medios por posición.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 12 de agosto de 2024	Fecha de fin: 15 de agosto de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa una vista que muestra un selector para seleccionar una unidad, una tabla que muestra el listado de posiciones de dicha unidad, una tabla que muestra los medios configurados para una posición seleccionada, además de las opciones para añadir un medio, modificar la configuración o eliminar una configuración.	

Tarea	
Número de tarea: 23	Número de historia de usuario: 11

Nombre de la tarea: Implementar la funcionalidad de añadir o modificar un medio a la configuración de una posición.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 15 de agosto de 2024	Fecha de fin: 21 de agosto de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permitirá añadir o modificar un medio a la configuración de una posición, así como la cantidad requerida de dicho medio.	

Tarea	
Número de tarea: 25	Número de historia de usuario: 14
Nombre de la tarea: Implementar la funcionalidad para actualizar la cantidad de cada medio disponible en una posición dada.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de inicio: 26 de agosto de 2024	Fecha de fin: 1 de septiembre de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permitirá mostrar un formulario con los medios disponibles cada posición y permitirá modificar la cantidad disponible de cada medio.	

Tarea	
Número de tarea: 27	Número de historia de usuario: 15
Nombre de la tarea: Implementar la funcionalidad registrar una incidencia	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 5 de septiembre de 2024	Fecha de fin: 12 de septiembre de 2024
Programador responsable: Zamir Alarcón Cabrera	
Descripción: Se implementa esta funcionalidad que permitirá mostrar un modal con tres pasos horizontales, en cada paso mostrará un formulario para obtener todos los datos necesarios para el registro de una incidencia.	