



Universidad de las Ciencias Informáticas
"Facultad 4"

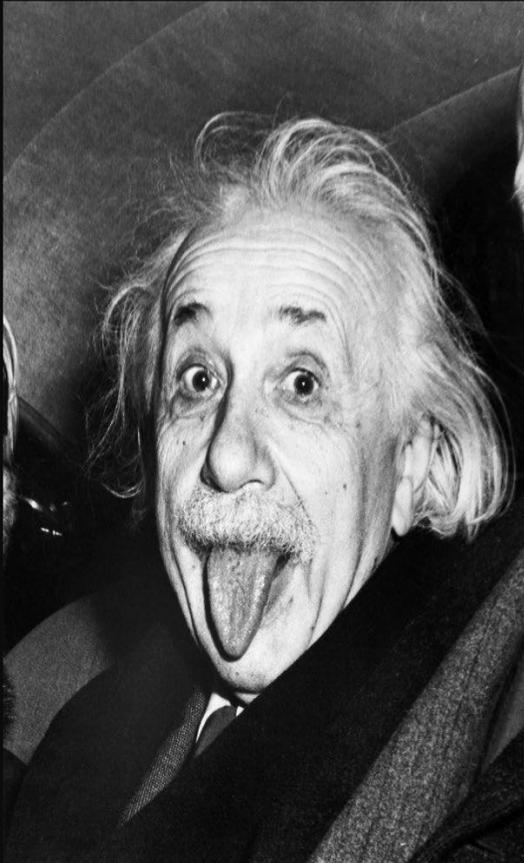
Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Título: " Sistema para la gestión de información asociada a la preparación para la defensa en la
Facultad 4 de la Universidad de las Ciencias Informáticas "

Autor: Frank Adivel Valero López
Tutor: "Ing. Yasirys Terry González"

"diciembre, 2022"

Frase:



**"If you want different results,
do not do the same things."**

•••

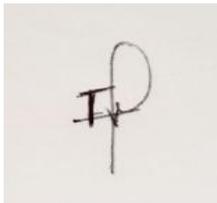
**"Si buscas resultados
distintos, no hagas
siempre lo mismo."**

ALBERT EINSTEIN

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis que tiene por título: “Sistema de gestión de información asociada a la preparación para la defensa en la Facultad 4 de la Universidad de las Ciencias Informáticas” y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los 8 días del mes de diciembre del año 2022.

<Frank Adivel Valero López>

A square image showing a handwritten signature in black ink on a light-colored background. The signature is stylized, starting with a large 'F' and ending with a vertical line.

Firma del Autor

<Yasirys Terry Gonzáles>

A square image showing a handwritten signature in blue ink on a light-colored background. The signature is cursive and includes the letters 'Y' and 'G'.

Firma del Tutor

RESUMEN

La Preparación para la Defensa en la disciplina encargada de transmitir a todos los estudiantes de la Universidad de las Ciencias Informáticas las habilidades militares para enfrentar cualquier situación excepcional que se presente en el país. Actualmente, los procesos vinculados a esta disciplina se gestionan de manera manual, causando insuficiencias en el proceso, evidenciadas en pérdidas de tiempo y un menor control de la información. El Sistema de gestión de Preparación para la Defensa se concibió como una herramienta informática para contribuir a mejorar la gestión de la información de todas las actividades desarrolladas en esta disciplina. Por tanto, el objetivo concreto de este trabajo fue desarrollar el análisis, diseño e implementación de funcionalidades como: gestión de personal en unidades militares y estructura del batallón, registro y control de la asistencia a las actividades, creación de informe resumen, la gestión del programa del bastión Universitario, control de reservistas en las diferentes unidades militares. Para guiar el desarrollo del sistema, se escogió como metodología de desarrollo Proceso Unificado Ágil en su versión para la Universidad de Ciencias Informáticas y para la selección de las herramientas se escogieron las que más características ventajosas presentaban para la consecución del resultado esperado. Entre las facilidades que brinda el sistema está: gestión y ubicación del personal; planificación, ejecución y evaluación de actividades; y generación de reportes mediante una interacción directa y fácil con los usuarios.

Palabras clave: control, gestión de información, Preparación para la Defensa, procesos, sistema.

ABSTRACT

The Preparation for Defense in the discipline in charge of transmitting to all the students of the University of Informatics Sciences the military skills to face any exceptional situation that arises in the country. Currently, the processes linked to this discipline are managed manually, causing insufficiencies in the process, evidenced in loss of time and less control of information. The Defense Preparedness Management System was conceived as a computer tool to help improve information management for all activities carried out in this discipline. Therefore, the specific objective of this work was to develop the analysis, design and implementation of functionalities such as: personnel management in military units and battalion structure, registration and control of assistance to activities, creation of summary reports, the management of the University bastion program, control of reservists in the different military units. To guide the development of the system, the Agile Unified Process was chosen as the development methodology in its version for the University of Informatics Sciences and for the selection of the tools the ones that presented the most advantageous characteristics were chosen to achieve the expected result. Among the facilities provided by the system are: personnel management and location; planning, execution and evaluation of activities; and report generation through direct and easy interaction with users.

Keywords: control, information management, Defense Preparation, processes, system.

Índice

Índice de tablas y figuras	IX
Introducción	1
Capítulo 1 Fundamentación Teórica	6
1.1 Antecedentes del Sistema de Defensa Nacional.....	6
1.2 La UCI y su sistema de Defensa	7
1.3 Gestión de la información	9
1.4 Sistemas de Gestión de la información	10
1.5 Sistemas de gestión de información de recursos humanos.....	11
1.6 Sistemas Homólogos	11
1.7 Metodología de desarrollo de software	16
1.8 Herramientas a utilizar en la propuesta de solución.....	20
Herramientas CASE (Computer Aided Software Environment)	20
Lenguaje de programación.....	21
Marco de Trabajo	22
Sistema Gestor de Base de Datos.....	23
Entorno de Desarrollo Integrado (IDE)	25
Servidor Web	26
1.9 Conclusiones parciales	26
Capítulo 2.....	28
2.1 Descripción de los procesos del negocio.....	28
2.2 Propuesta de solución	29
2.3 Definición de los actores del sistema	30
2.4 Captura de Requisitos.....	30
2.5 Patrón Arquitectónico.....	47
2.6 Patrones de Diseño	49
2.7 Modelo del diseño	51
2.8 Modelo de implementación	54
2.9 Diagrama de despliegue	55
2.10 Conclusiones parciales.....	56
Capítulo 3.....	58

3.1	Fase de Pruebas	58
3.2	Estrategia de pruebas.....	58
3.3	Pruebas de aceptación	62
3.3	Aplicación de pruebas de aceptación	63
3.4	Conclusiones parciales	68
	Referencias bibliográficas.....	72
	Anexos	79

Índice de tablas y figuras

Ilustración 1: Estructura de Defensa de la UCI (Elaboración propia).....	8
Ilustración 2: Fases de la metodología AUP-UCI(UCI, 2014).	19
Ilustración 3: Funcionamiento del patrón arquitectónico MVC	48
Ilustración 4: Ejemplo de los patrones en el código del sistema	51
Ilustración 5: Diagrama de Clases (Elaboración propia)	52
Ilustración 6: Diagrama Entidad-Relación.....	53
Ilustración 7: Diagrama de componentes (Elaboración propia)	54
Ilustración 8: Diagrama de despliegue (Elaboración propia).....	55
Ilustración 9: Clasificación de las no conformidades para una primera iteración de desarrollo (18 requisitos funcionales.)	67
Ilustración 10: Clasificación de las no conformidades para la segunda iteración de desarrollo (10 requisitos funcionales.)	68
Ilustración 11: Clasificación de las no conformidades para la tercera iteración de desarrollo (8 requisitos funcionales.)	68
Tabla 1: Comparación entre aplicaciones de escritorio y aplicaciones web (Elaboración propia)	15
Tabla 2: Comparación entre las metodologías tradicionales y las metodologías ágiles(Kuz, Falco, & Giandini, 2020).	17
Tabla 3: Actores que intervienen en el sistema (Elaboración propia)	30
Tabla 4: Requisitos funcionales identificados	31
Tabla 5: Requisitos no funcionales identificados	42
Tabla 7: HU Incluir usuario	45
Tabla 22: Clases de equivalencia	63
Tabla 23: Diseño de caso de prueba incluir actividad para el aseguramiento del bastión	64
Tabla 27: Cantidad de no conformidades según el tipo de error y riesgo detectado	66
Tabla 6: HU: Autenticar usuario	79
Tabla 8: HU: Modificar datos del usuario	79
Tabla 9: HU: Eliminar usuario	81
Tabla 10: HU: Listar usuarios	83
Tabla 11: HU: Filtrar usuarios	84
Tabla 12: HU: Crear actividades para el aseguramiento del bastión	85
Tabla 13: HU: Modificar actividades para el aseguramiento del bastión.....	87
Tabla 14: HU: Eliminar actividades para el aseguramiento del bastión	89
Tabla 15: HU: Consultar datos de unidad militar	91
Tabla 16: HU: Generar reporte de informe resumen de la actividad del bastión	92
Tabla 24 : Diseño de caso de caso de prueba para RF Eliminar actividad para el aseguramiento del bastión	94
Tabla 25: Diseño de caso de prueba para acción Seleccionar acciones en lote	94
Tabla 26: Diseño de caso de prueba para RF Modificar actividades para el aseguramiento del bastión	95

Introducción

La preparación militar para la defensa en Cuba forma parte de un proceso que surge a raíz de la posibilidad real de que se produzca un ataque militar contra Cuba. El Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR), como institución militar del Estado Cubano, ha considerado como elemento clave: el empleo del capital humano, en interés de asignarle a cada ciudadano un lugar en la defensa. Esta institución tiene como prioridad el sistema defensivo territorial y el perfeccionamiento de las condiciones físicas, psíquicas, políticas y sociales existentes. Además de la preparación militar y otras cualidades personales de modo que cada efectivo pueda ser empleado apropiadamente, de acuerdo con las misiones que le sean asignadas (Minfar, 2022).

En el ámbito universitario, se ha implementado y aprobado como parte del currículo de las diferentes carreras de pregrado, la disciplina docente: Preparación para la Defensa (PPD). Dicha disciplina abarca contenidos teóricos e histórico-militares, dirigidos a desarrollar los conocimientos necesarios sobre defensa y seguridad nacional de los futuros profesionales que una vez graduados según la rama en la que se desempeñen se mantengan fieles a los intereses del pueblo, capaces de materializar las exigencias de la concepción estratégica de Guerra de Todo el Pueblo. La PPD se imparte mediante diferentes asignaturas que divergen en dependencia de las condiciones y características de cada institución; y en función de la carrera que se estudie (Peña Galbán, 2018).

La Universidad de las Ciencias Informáticas (UCI) se encuentra dentro de los centros que pertenecen al Ministerio de Educación Superior (MES), y en ella se aplica también la estrategia de formación militar, tanto para estudiantes como trabajadores. En el pregrado se gestiona la preparación teórica mediante la PPD, consolidándose en las asignaturas Seguridad Nacional y Defensa Nacional, y de forma práctica se convoca a ejercicios y actividades como: Días de la Defensa Nacional, Bastiones Universitarios, Meteoros, entre otras. Todas con el objetivo fundamental de tributar a la preparación de la defensa del país y mantener su soberanía.

La UCI, al contar con un gran volumen de personal se hace imprescindible mantener activo su Sistema de Defensa, para ello es necesario que cada área de la universidad gestione de manera eficiente la información que se genera de todas actividades antes mencionadas. La Facultad 4

como una de las áreas que integra el Sistema de Defensa de la UCI, al comienzo de cada curso escolar mediante el Vicedecanato de Extensión y Residencia (VDER) en estrecha relación con el Área de Atención Militar UCI, gestiona la información referente a la preparación para la defensa. Se reconoce un alto nivel de compromiso y responsabilidad ante esta tarea, no obstante, se identifica como insuficiencia que no se realiza una adecuada gestión de la información debido a que:

- ✓ Se manejan grandes volúmenes de información, la cual es procesada de manera manual, lo que trae consigo un mayor tiempo y esfuerzo para llevar a cabo este proceso.
- ✓ Existen demoras en la ejecución de los procesos, los datos son almacenados en formato duro y múltiples archivos de ofimática, por lo que se encuentran sujetos a la pérdida, el deterioro y duplicación de la información.
- ✓ Existen deficiencias en la distribución y ubicación del personal, asociadas a la carencia de planillas digitales que permitan aglutinar los datos personales y militares de cada efectivo, lo que dificulta la disponibilidad del batallón ante cualquier evento y unido a esto el cumplimiento de sus tareas y misiones.
- ✓ Se dificulta la interacción con la información generada del proceso, lo que limita el acceso y actualización permanente de los datos, así como la manipulación y divulgación de los resultados.

Teniendo en cuenta la situación problemática antes descrita, se identifica como **problema de la investigación**: ¿Cómo contribuir a la gestión de la información asociada a la preparación para la defensa en la Facultad 4 de la Universidad de las Ciencias Informáticas? Se definió como **objeto de estudio**: La gestión de información.

Para dar solución al problema de investigación planteado, se define como **objetivo general**: Desarrollar un sistema para la gestión de información asociada a la preparación para la defensa en la Facultad 4 de la Universidad de las Ciencias Informáticas.

Dentro del objeto de estudio de la investigación se precisa como **campo de acción**: La gestión de la información asociada a la preparación para la defensa.

Para dar cumplimiento al objetivo general, se plantean los siguientes objetivos específicos:

- ✓ Establecer los referentes teóricos de la investigación mediante un estudio sobre la preparación para la defensa en la UCI, su sistema de defensa y los sistemas de gestión de la información.
- ✓ Identificar tendencias y enfoques de herramientas en el proceso de desarrollo de software, para comparar sus características e identificar ventajas y limitaciones.
- ✓ Diseñar el sistema informático a desarrollar, con la utilización de buenas prácticas definidas para procesos de desarrollo de software.
- ✓ Implementar un sistema informático que permita la gestión de la información referente a la preparación para la defensa en la Facultad 4 de la UCI.
- ✓ Validar el sistema informático propuesto a partir de diversas métricas de pruebas de *software*, así como la realización de pruebas de funcionalidad y pruebas unitarias.

Para dar cumplimiento a los objetivos específicos propuestos anteriormente, se utilizaron los siguientes métodos científicos de investigación teóricos y empíricos. Dentro de los métodos teóricos se utilizaron:

- ✓ Histórico-Lógico: permitió realizar un estudio sobre la estructura del Sistema de defensa en la UCI, los sistemas y herramientas para la gestión de la información existentes, las tendencias del uso actual de las mismas, con el fin de seleccionar los patrones más apropiados para implementar en la propuesta de solución y darle cumplimiento al objetivo general de la presente investigación.

- ✓ Analítico-Sintético: permitió realizar el estudio teórico de la investigación facilitando el análisis de documentos y la extracción de los elementos más importantes acerca del proceso de desarrollo de sistemas de gestión de la información.
- ✓ Inductivo-Deductivo: fue utilizado para el razonamiento de la información consultada y llegar a un grupo de conocimientos particulares y generales sobre los procesos de gestión de información.

Dentro de los métodos de investigación empíricos se emplearon:

- ✓ Entrevista: se aplicó para comprobar la forma en que se gestiona la información referente a la defensa, propiciando el levantamiento de requisitos del sistema y obtención de información necesaria para el desarrollo de la investigación.
- ✓ Observación: se empleó a través de la realización del estudio de los elementos que componen la gestión de la información de la defensa existente en la UCI, además en el análisis de algunas herramientas de gestión de información a nivel mundial y en Cuba, que brindan información sobre el proceso de gestión y seguridad de la información que poseen, así como la observación de la ejecución de actividades realizadas en la UCI tales como: el Bastión Universitario.

Estructura del presente trabajo:

Capítulo 1: Fundamentación teórica

En este capítulo, se muestran los elementos teóricos necesarios para la comprensión de la investigación, incluyendo el establecimiento del estado del arte a través del estudio de herramientas de gestión de información a nivel mundial y nacional. Se analiza y selecciona la metodología de desarrollo en función de las necesidades y las características del equipo de desarrollo.

Capítulo 2: Propuesta de solución

En este capítulo, se realiza una breve descripción de los principales procesos del negocio y el análisis de la propuesta de solución. Luego se definen los actores que intervienen en el sistema, así como los requisitos funcionales, no funcionales y la descripción de las principales historias de usuario. Además, se muestra la arquitectura a la cual responde el sistema y la evidencia de los patrones de diseño utilizados. Se abordan aspectos significativos de la implementación del sistema, así como los artefactos generados del proceso

Capítulo 3: Pruebas

Se muestran los resultados de la validación de los requisitos y el diseño, a través de la aplicación de técnicas y métricas respectivamente. Además, se hace un resumen de los resultados obtenidos tras la aplicación de las pruebas de funcionamiento y unitarias realizadas a la al sistema.

Capítulo 1 Fundamentación Teórica

En el presente capítulo se exponen conceptos que permitirán comprender la esencia de la preparación para la defensa en la UCI, como negocio a informatizar, además, se explican elementos teóricos que facilitan la comprensión del proceso de Gestión de información, como objeto de la investigación. Se abordan tendencias asociadas al desarrollo y utilización de los sistemas de gestión de información a nivel internacional y nacional. También se describen las principales características y funcionalidades de las tecnologías a emplear en la construcción de la propuesta de solución.

1.1 Antecedentes del Sistema de Defensa Nacional

La disciplina PPD surge como parte de un proceso de formación y preparación militar de la sociedad cubana. Luego al aprobarse la Directiva 29 del MINFAR se planteó perfeccionar el sistema de preparación para la defensa de los estudiantes universitarios, al firmarse un convenio entre el MINFAR y el MINED en septiembre de 1975. Esta disciplina tenía una proyección específica que vinculaba las exigencias de los futuros egresados, sus responsabilidades y acciones en la defensa. Por ello, su principal misión es, asesorar, preparar las estructuras de dirección y metodológicas del MINED para el desarrollo de la Educación Patriótica de Preparación para la Defensa de niños, adolescentes y jóvenes matriculados en el Sistema nacional de Educación en coordinación con las educaciones, los organismos y organizaciones implicadas(MINED, 2021).

El Sistema de Defensa Nacional cubano tiene como tarea primordial mantener las conquistas alcanzadas por la Revolución, por lo que todo el esfuerzo militar se dedica estrictamente a la defensa del país. Este se constituye y prepara desde tiempo de paz, para dirigir el país en las condiciones de: estado de guerra, guerra, movilización general o estado de emergencia. Mientras que el MINFAR es la institución que dirige, ejecuta y controla la preparación de los ciudadano(a)s para la defensa; esto involucra la protección de todo el territorio nacional, incluidos el mar territorial y el espacio aéreo que sobre estos se extiende; la preparación y realización de la lucha armada; y

la contratación, adquisición, producción y uso del material de guerra que satisfaga las necesidades de la defensa(Bunch, 2011).

El autor de esta investigación concuerda en que las universidades cubanas deben integrarse de manera eficiente a las actividades de preparación para la defensa. Para ello se debe mantener el adecuado control de su estructura militar y la actualización constante de su personal, facilitando así, la búsqueda de estrategias que tributen a mejorar la capacitación, preparación y gestión de información referente a la defensa.

1.2 La UCI y su sistema de Defensa

La UCI al ser una universidad con características especiales y albergar un gran número de capital humano, cuenta con un sistema de defensa propio, compuesto por la Dirección de Seguridad, Protección y Defensa (DSPD), la Cátedra de PPD, Área de Atención Militar y los Vicedecanatos de Extensión y Residencia (VDER de las facultades; dirigido por el Consejo de Defensa UCI. A cada una de las áreas que integran el Sistema de Defensa de la UCI se le asignan misiones específicas, donde el DSPD se encarga del diseño, planificación y realización de las actividades de preparación militar en coordinación con la Cátedra de PPD.

Sistema de Defensa UCI



Ilustración 1: Estructura de Defensa de la UCI (Elaboración propia)

Como se muestra en la Ilustración 1, cada una de las áreas que integran el Sistema de Defensa UCI, tienen misiones específicas, donde se destaca la DSPD encargada del diseño, planificación y realización de las actividades de preparación militar en la universidad, en previa coordinación con la Cátedra de PPD. Los VDER de cada una de las facultades se encargan de gestionar toda la información referente a las estructuras militares y las actividades de defensa de sus batallones, entre las que destacan:

- ✓ Realizar la distribución del personal en las estructuras de sus batallones y la asignación de actividades.
- ✓ Mantener actualizada la información, además de disponible para la movilización oportuna de los batallones ante situaciones de carácter excepcional.
- ✓ Archivar la ubicación de cada efectivo y los resultados del control y evaluación de las

actividades ejecutadas.

- ✓ Mantener la confidencialidad sobre los datos almacenados, mediante un acceso limitado a los mismos.

En caso de realizarse una valoración del Sistema de Defensa de la UCI, su estructura, misiones, responsables y nivel de organización para con cada efectivo en la defensa. No cabe duda que cada área y específicamente las facultades por ser las que aglutinan mayor cantidad de personal entre estudiantes y trabajadores; tienen una gran responsabilidad en la gestión de la información que generan, pues de la calidad y pertinencia de esta, depende la efectividad del cumplimiento de las tareas y misiones de cada uno de los batallones.

1.3 Gestión de la información

Gestión de la información (GI) es la denominación convencional de un conjunto de procesos por los cuales se controla el ciclo de vida de la información, desde su obtención (por creación o captura), hasta su disposición final (su archivo o eliminación). Tales procesos también comprenden la extracción, combinación, depuración y distribución de la información a los interesados. El objetivo de la gestión de la información es garantizar la integridad, disponibilidad y confidencialidad de la información(Fernández, 2017).

En el contexto de las organizaciones, la gestión de la información se puede identificar como la disciplina que se encargaría de todo lo relacionado con la obtención de la información adecuada, en la forma correcta, para la persona indicada, en el momento oportuno, en el lugar apropiado y articulando todas las operaciones para el desarrollo de una acción correcta. En este contexto, los objetivos principales de la Gestión de la Información son: aumentar el valor y los beneficios derivados del uso de la información, disminuir el coste de adquisición, procesamiento y uso de la información, determinar responsabilidades para el uso efectivo, eficiente y económico de la información y asegurar un suministro continuo de la información(Fernández, 2017).

Urdaneta plantea: «la gestión de información es todo lo que tiene que ver con obtener la información correcta, en la forma adecuada, para la persona indicada, al costo correcto, en el momento oportuno, en el lugar indicado para tomar la acción precisa.»(Urdaneta, 1985)

White (1985) la denomina como: «la coordinación eficiente y eficaz de la información procedente de fuentes internas y externas(Urdaneta, 1990).

De lo anteriormente planteado, el autor de esta investigación asume la posición de Davis y Olson, producto que, al gestionar la información de manera correcta se, sostienen las funciones operativas. Este proceso ayuda a tomar las mejores decisiones, así como determinar responsabilidades para su uso efectivo, eficiente y económico.

1.4 Sistemas de Gestión de la información

Los sistemas de gestión de información son un tipo de sistemas de información que recopilan y procesan información de diferentes fuentes para ayudar en la toma de decisiones en lo referente a la gestión de la organización(Kyocera, 2021). Los sistemas de gestión de información proporcionan información en forma de informes y estadísticas. El siguiente nivel en la jerarquía organizacional está ocupado por gerentes y supervisores de bajo nivel. Este nivel contiene los sistemas informáticos que están destinados a ayudar a la gestión operativa en la supervisión y control de las actividades de procesamiento de transacciones que se producen a nivel administrativo. Los sistemas de gestión de información son los tipos de sistemas de información que toman los datos internos del sistema y los resumen en formatos útiles como informes de gestión para utilizarlos como apoyo a las actividades de gestión y la toma de decisiones.

Algunos autores como Davis y Olson (1985) conceptualizan los sistemas de gestión de información como un: «sistema integrado y automatizado para proveer la información que sostenga las funciones de operatividad, gestión y toma de decisiones en una organización»(Urdaneta, 1990).

Por otra parte, Gonzáles lo define como “el conjunto de políticas y normas relacionadas entre sí que se establecen para el acceso y tratamiento de los recursos de información, incluye los registros

administrativos y los archivos, el soporte tecnológico de los recursos y el público a que se destina. En su evolución el sistema puede manejar la función de inteligencia corporativa y generar productos de inteligencia”(Urdaneta, 1990).

El autor cuando analiza en detalle las funciones de los sistemas de gestión de información, considera que estos deben aprovechar al máximo sus recursos de información. Proceso que se realiza en función de la mejora continua y la toma de decisiones organizacional a todos los niveles jerárquicos, desde la cúspide estratégica hasta la base operativa.

1.5 Sistemas de gestión de información de recursos humanos

Un Sistema de Información de Recursos Humanos, o también denominado SIRH, reúne todas las herramientas de recursos humanos que necesita el departamento de personal para realizar su actividad del día a día con la máxima eficiencia. Por tanto, se puede definir un SIRH como un conjunto de softwares que las organizaciones utilizan para administrar las funciones internas de RR. HH. Es capaz de simplificar y acelerar los flujos de información dentro de la empresa pudiendo gestionar mejor las competencias de los empleados y obtener mejores resultados en cuanto a rendimiento. Esto, por supuesto, repercute directamente en los resultados del negocio(Kenjo, 2022).

Se puede definir un SGIRH a partir de lo antes planteado: como una herramienta informática que, a partir del uso adecuado de la información almacenada de las personas, logre una gestión eficiente, en función del cumplimiento de los objetivos específicos de la organización. En el caso de la presente investigación sería: un sistema informático que, a partir de las políticas, estatutos, indicaciones y documentos rectores del proceso de la defensa, realice la gestión eficiente de la información referente a la estructura militar y las actividades de la defensa de la facultad 4.

1.6 Sistemas Homólogos

Teniendo en cuenta las particularidades de los sistemas que gestionan información militar, su confidencialidad y política reservada, se realiza un estudio de sistemas que, si bien no gestionan este tipo de información, tributarán en la obtención de funcionalidades y algunos aspectos similares

que se podrán utilizar en el desarrollo de la aplicación web. A continuación, se detalla un estudio sobre algunos sistemas internacionales y nacionales sirven de guía y constituirán la base de la investigación.

Sentrifugo

El producto Sentrifugo provee soluciones avanzada de gestión del capital humano ayudando a las organizaciones a mejorar, gestionar, recompensar y retener sus recursos más importantes: las personas. Sentrifugo es un potente sistema de recursos humanos online, equipado con excelentes funcionalidades que reducen las tareas administrativas, dinamiza la gestión de las personas e involucra a los empleados de forma más productiva. Con una navegación intuitiva, pantallas fáciles de usar, robusta seguridad, sistema de alertas y notificaciones, flexibilidad en la elaboración de reportes y sistema de autoservicio integrado para directivos y empleados. Ofrece además un entorno ágil y cómodo para la gestión del personal en una empresa.

Este producto facilita la recopilación de información sobre los empleados. De forma centralizada, incluye datos personales y profesionales, historial laboral, habilidades y calificaciones, expedientes disciplinarios y formación profesional de cada persona vinculada al negocio. Reflejando toda la información sobre la gestión de las personas en el sistema de recursos humanos, de forma sencilla y de fácil acceso; muy similar a un perfil personal y psicológico.

Sistema de Gestión de Personal Militar

Este sistema realizado por la Universidad Pontificia Comillas de Madrid, es una aplicación web válida para cualquier ejército en su país, ya sea de tierra o de aire. Está destinada a mejorar el sistema actual de la Jefatura de Personal y permite una mayor rapidez y comodidad mediante la gestión del personal militar. Mantiene un historial de la carrera militar y la trayectoria profesional de todo el personal militar. La aplicación también permite realizar búsquedas de personal según determinados datos introducidos, facilita la realización de plantillas orgánicas según la unidad seleccionada, ya sea una fragata, una corbeta o un portaaviones.

El sistema lleva una gestión integral de la información del personal militar, en todo lo que se refiere a altas, bajas o modificaciones. Permite mostrar la estructura orgánica de las Fuerzas Armadas basada en el ordenamiento jerárquico de sus miembros, de forma que se logre la adecuada comprensión de los diferentes niveles de mando dentro de cada estructura.

Sistemas nacionales de gestión de información

En el ámbito nacional, también se han desarrollado diversos sistemas asociados a la gestión de información. Por lo que se describen brevemente a continuación los que más relación tienen con el propósito de esta investigación:

Sistema de Gestión Universitaria (XAUCE SGU)

El Sistema de Gestión Universitaria (SGU), desarrollado en el Centro de Informatización Universitaria (CENIA), está diseñado para ser una solución integral en la gestión de los procesos sustantivos de la universidad, incluyendo los recursos humanos asociados a los mismos. El SGU gestiona información de las áreas de procesos: Pregrado, Postgrado, Desarrollo, Tecnologías, Investigación, Ingreso, Ubicación laboral, Residencia, Extensión universitaria, Cooperación Internacional, Biblioteca y Teleformación.

Sistema Integral de Gestión de Recursos Humanos (GREHU)

Desarrollada por el Centro de Estudios de Ingeniería en Sistemas (CEIS) de la Ciudad Universitaria José Antonio Echeverría (CUJAE), constituye una herramienta para la gestión integral de información de recursos humanos. Esta propone procedimientos y pautas a seguir para un mejor control y manipulación de los datos contenidos en el modelo principal del expediente laboral de los trabajadores, las actas de sanciones y medidas disciplinarias, entre otros, relacionados con la nueva concepción del Inventario de Personal. Permite realizar preselección tanto de candidatos externos como de los trabajadores existentes según criterios preestablecidos para realizar promociones o nuevos contratos, basados en el grado de prioridad según el análisis de la información almacenada.

Después del estudio de los SGIRH antes mencionados, se concluye que ninguno se adecua a las especificidades del negocio inherente a la investigación. Esto se debe a que fueron creados con otras finalidades, lo cual evidencia la necesidad de desarrollar una solución propia a la problemática planteada referente a la gestión de información de preparación para la defensa en la Facultad 4.

Aunque es preciso reconocer que el estudio de dichas soluciones aportó elementos significativos a tomar en consideración para el desarrollo de la propuesta de solución como es el caso de:

- Creación de la estructura de los formularios de perfil personal. (Presente en los sistemas Sentrifugo y Sistema de Gestión de Personal Militar y Plantillas Orgánicas).
- Búsqueda de personal según determinados criterios. (Presente en el Sistema Integral de Gestión de Recursos Humanos (GREHU)).
- Almacenamiento de la documentación. (Presente en el sistema Sentrifugo).
- Mostrar la estructura basada en el ordenamiento jerárquico. (Presente en el Sistema de Gestión de Personal Militar y Plantillas Orgánicas).

Luego de conceptualizar los principales elementos de la gestión de información y establecer un análisis de herramientas similares existentes, lo cual respalda la necesidad de implementar un sistema de gestión de información acorde a las necesidades. Se hace necesario decidir el camino hacia la solución, mediante el estudio de la factibilidad de implementación en caso de ser una aplicación web o de escritorio, en consecuencia, con las necesidades planteadas.

A continuación, se muestra una tabla comparativa entre las aplicaciones de escritorio y las aplicaciones web, atendiendo a una serie de características:

Tabla 1: Comparación entre aplicaciones de escritorio y aplicaciones web (Elaboración propia)

Características	Aplicación de escritorio	Aplicación web
Portabilidad	Puede presentar incompatibilidades con el sistema operativo, así como dependencia de los recursos que posea la computadora.	El sistema puede ser usado a través de un navegador, sin necesitar grandes prestaciones de recursos en la computadora.
Accesibilidad	Solo en la computadora donde se haya instalado previamente la aplicación	Cualquier lugar con acceso a la red donde se encuentre montado el servidor.
Actualización y soporte	Los cambios hay que realizarlos en cada computadora donde se encuentre instalada la aplicación.	Es suficiente con realizar los cambios en el servidor WEB.
Infraestructura y movilidad	Está restringido a la ubicación de la computadora local.	Solo se tiene que conectar a la red donde se encuentre montado el servidor.

A partir de la comparación establecida anteriormente y en correspondencia con las necesidades existentes; el autor de este trabajo determina desarrollar la propuesta utilizando aplicaciones web, lo cual brinda múltiples beneficios en cuanto a portabilidad, actualización, soporte e infraestructura.

Para proseguir, se hace necesaria la selección de la metodología de desarrollo de software, y las herramientas y tecnologías a emplear durante el desarrollo de la propuesta.

1.7 Metodología de desarrollo de software

Una metodología de desarrollo de software es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo. Las metodologías se basan en una combinación de los modelos de proceso genéricos. Definen artefactos, roles y actividades, junto con prácticas y técnicas recomendadas. La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Esta comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta cumplir el objetivo por el cual fue creado (Pacienza, 2018).

En la actualidad existen diversas metodologías de desarrollo de software para guiar a los equipos de trabajo en la creación de un nuevo software, atendiendo a su complejidad son separadas en dos grupos: metodologías tradicionales o formales y las ágiles o ligeras. **Las metodologías tradicionales** se caracterizan por tener una estructura de desarrollo claramente establecida, lineal y poco flexible ante cambios de un entorno volátil. Presentan un alto costo al ser implementadas. Tienen roles, actividades y artefactos claramente definidos. Requieren de una documentación exhaustiva, clara y detallada de todo el proyecto. Y centran su atención en cumplir un plan establecido desde el inicio del desarrollo, por todas las partes involucradas (Esan, 2021).

Las metodologías ágiles de desarrollo de software buscan proporcionar en poco tiempo pequeños componentes de software en funcionamiento para aumentar la satisfacción del cliente. Estas metodologías utilizan enfoques flexibles y el trabajo en equipo para ofrecer mejoras constantes. Por lo general, el desarrollo ágil de software implica que pequeños equipos autoorganizados de desarrolladores y representantes empresariales se reúnan regularmente en persona durante el ciclo de vida del desarrollo de software. La metodología ágil favorece un

enfoque sencillo de la documentación de software y acepta los cambios que puedan surgir en las diferentes etapas del ciclo de vida, en lugar de resistirse a ellos(Esan, 2021).

A continuación, se muestra una tabla con una serie de criterios comparativos entre estas metodologías:

Tabla 2: Comparación entre las metodologías tradicionales y las metodologías ágiles(Kuz, Falco, & Giandini, 2020).

Metodologías Tradicionales	Metodologías Ágiles
Enfoque Predictivo	Enfoque Adaptativo
Orientado a procesos	Orientado a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de productos de software
Documentación extensa	Poca documentación

Hay una gran diversidad de metodologías, aunque todas ellas caen dentro de alguna de las dos clasificaciones mencionadas: ágiles o tradicionales. Sin embargo, existe una nueva categoría: las metodologías híbridas.

Las metodologías híbridas pretenden retomar las ventajas de las metodologías existentes, de tal forma que son una combinación de las mejores prácticas descritas en cada una de ellas. A partir de lo descrito, teniendo en cuenta las necesidades y condiciones presentes para la realización de la propuesta de solución, donde:

- El tiempo de desarrollo es acotado.
- El cliente es parte del equipo de desarrollo.
- El equipo de desarrollo es pequeño.
- Se requiere entregas constantes de versiones del producto.
- El negocio no requiere grandes volúmenes de documentación.

La Universidad de las Ciencias Informáticas (UCI) desarrolló una versión de la metodología de desarrollo de software AUP (Proceso Ágil Unificado), con el fin de crear una metodología que se adapte al ciclo de vida definido por la actividad productiva de la universidad. Esta versión decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma y se unifican las restantes fases de la metodología de desarrollo de software AUP en una sola, nombrada Ejecución y agregándose también una nueva fase llamada Cierre (Ecured, 2015).

La variación de AUP describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles como:

- Desarrollo dirigido por pruebas.
- Modelado ágil.

Se basa en tres fases como se muestra en la figura:



Ilustración 2: Fases de la metodología AUP-UCI(UCI, 2014).

Se decide seleccionar AUP-UCI debido a que se centra en equipos de desarrollo pequeños. Además, brinda una gestión de cambios ágil que hace que el equipo de desarrollo se adapte a nuevas condiciones, que en este caso los requisitos pueden cambiar a decisión del cliente. Otro punto a su favor, es la forma en que se planifica el proyecto y realiza la estimación del tiempo, pues se cuenta con poco tiempo para el desarrollo de la aplicación. Finalmente, como factor más determinante de esta selección es que esta es la metodología utilizada actualmente en la UCI.

Para dirigir el desarrollo del sistema de gestión de preparación para la defensa de la facultad 4, se empleará el escenario 4 de la metodología AUP variante UCI. La misma se basa en la descripción de historias de usuario, debido a que se aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido donde el cliente estará siempre acompañando al equipo de desarrollo. Para elaborar HU los integrantes del proyecto debe establecer conversaciones acerca de las necesidades de los clientes. Se recomienda en proyectos no muy extensos, pues una HU no debe poseer demasiada información.

El dominio del proyecto a desarrollar para dar solución al problema a resolver identificado en el presente trabajo no se considera altamente complejo, por tanto, puede ser descrito a través de HU.

La Vicedecana de Extensión Universitaria de la Facultad 4, como representante de los intereses del cliente estará en constante comunicación con el equipo de desarrollo, por lo que contribuirá a la confección de las HU. Además, con la selección del escenario 4 de la metodología se garantizará un alto nivel de asimilación de cambios porque las HU se caracterizan por ser dinámicas y flexibles y posibilitan reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas.

1.8 Herramientas a utilizar en la propuesta de solución

Para definir las herramientas y tecnologías a utilizar durante el desarrollo de la propuesta, se realizó un estudio sobre las líneas de desarrollo utilizadas en la facultad 4 para aplicaciones web de gestión. Con el fin de determinar cuál de las líneas se adecúa mejor a las necesidades y especificidades del problema planteado, se describen a continuación los principales elementos que las caracterizan, y las herramientas y tecnologías que proponen en cada caso

Herramientas CASE (Computer Aided Software Environment)

Las herramientas CASE son diversas aplicaciones informáticas o programas informáticos destinadas a aumentar el balance en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras (GlosarioIT, 2022). Es una de las herramientas UML CASE del mercado, considerada como muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones.

Visual Paradigm 8.0

Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de las clases. Fue creada para el ciclo vital completo del

desarrollo de software que lo automatiza y acelera, lo que permite la captura de requisitos, análisis, diseño e implementación.

Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación.

Lenguaje de programación

Un lenguaje de programación es un lenguaje formal (o artificial, es decir, un lenguaje con reglas gramaticales bien definidas) que le proporciona a una persona, en este caso el programador, la capacidad de escribir (o programar) una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico o lógico de un sistema informático, de manera que se puedan obtener diversas clases de datos o ejecutar determinadas tareas. A todo este conjunto de órdenes escritas mediante un lenguaje de programación se le denomina programa informático(Equipo Editorial, 2021).

Python: Es uno de los lenguajes de programación más usados actualmente y su uso sigue creciendo. Posee unas características muy potentes: es de código abierto, tiene una sintaxis sencilla y es fácil de entender, por lo que ahorra tiempo y recursos(KeepCoding, 2021).

Java: Es un lenguaje de propósito general, orientado a objetos y diseñado para tener las dependencias de implementación mínimas posibles. Con Java se pueden crear aplicaciones y procesos en múltiples dispositivos, además, su ámbito de acción es muy amplio, por lo que permite crear software para dispositivos móviles, terminales de venta, además de páginas web(KeepCoding, 2021).

C++: se caracteriza por ser extremadamente rápido y estable, lo que lo sitúa como uno de los mejores lenguajes de programación y, por tanto, también uno de los lenguajes de programación más usados. Además, proporciona STL (biblioteca de plantillas estándar), es decir, un conjunto de bibliotecas listas para usar para diferentes estructuras de datos, operaciones aritméticas y algoritmos(KeepCoding, 2021).

Se escoge Java como lenguaje de programación por su robustez (fiable). Java pone mucho énfasis en la comprobación temprana de todos los posibles errores y excepciones. Como parte del manejo de excepciones en Java, el compilador puede llegar a confirmar todas las posibilidades en situaciones de tiempo de ejecución, lo que da mucha fiabilidad a los clientes. Además, permite acceder a las aplicaciones basadas en la web desde cualquier lugar. Los desarrolladores de Java prueban estas aplicaciones para todos los navegadores web posibles, lo que hace que puedan ejecutarse fácilmente en cualquier sistema operativo posible como Windows, Mac y Linux.

Marco de Trabajo

Marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. Es decir; un marco de trabajo web se puede definir como un conjunto de componentes (por ejemplo: clases, descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web y es precisamente el que se utiliza en estos estudios(UNIR, 2021). Tienen como objetivo principal ofrecer una funcionalidad definida, auto contenida, siendo construidos usando patrones de diseño, y su característica principal es su alta cohesión y bajo acoplamiento.

Vaadin 8.0: Vaadin es una plataforma de desarrollo para Java. Vaadin incluye un conjunto de Componentes Web, un *framework* para aplicaciones web y un conjunto de herramientas que permite a los desarrolladores implementar interfaces gráficas de usuario usando exclusivamente el lenguaje de El modelo de programación de Vaadin Flow permite a los desarrolladores usar Java como lenguaje de programación para la implementar Interfaces de Usuario (IU) sin tener que usar

HTML o JavaScript directamente. Vaadin Flow ofrece una arquitectura del lado del servidor lo que significa que la mayor parte de la lógica UI es ejecutada en el servidor reduciendo la exposición de ataques de seguridad. En el lado del cliente, Vaadin Flow está construido sobre los estándares de Componentes Web. La comunicación cliente-servidor es manejada automáticamente por medio de WebSocket o HTTP con mensajes JSON ligeros que actualizan ambos, la IU en el navegador y el estado en el servidor(Michael «Monty», 2022).

Spring: es un *framework* de código abierto para la plataforma Java, es seguramente el más popular y utilizado del mundo. Se basa en la arquitectura MVC (Modelo-Vista-Controlador), además tiene buenas características de seguridad que puede llamar como funciones ya escritas. Esto hace que procesos como la autenticación, la verificación y la validación sean mucho más fáciles de incluir en cualquier proyecto(Muradas,2018).

Se decide escoger Vaadin como *framework* de desarrollo ya que el equipo de desarrollo posee dominio previo de dicho marco de trabajo. Además, cumple con las características necesarias para el desarrollo de la propuesta solución entre las cuales se destacan: que es un *framework open source*, utiliza java como único lenguaje de programación, es de agradable presentación al usuario y es soportado por la mayoría de navegadores webs conocidos sin necesidad de *plugins*.

Mientras que, para el desarrollo de la aplicación del lado del servidor, se seleccionó Spring, el cual permite crear aplicaciones auto contenidas, con esto se enfoca en el desarrollo, dejando a un lado la arquitectura y delegando a *Spring Boot* labores de: configuración de dependencias y el despliegue del servicio o aplicación a un servidor de aplicaciones.

Sistema Gestor de Base de Datos

Un sistema gestor de base de datos (SGBD) o *Database Management System* (DBMS) es un conjunto de programas invisibles para el usuario final con el que se administra y gestiona la información que incluye una base de datos. Los gestores de datos o gestores de base de datos permiten administrar todo acceso a la base de datos, pues tienen el objetivo de servir de interfaz entre esta, el usuario y las aplicaciones. Entre sus funciones se encuentran la de permitir a los

usuarios de negocio almacenar la información, modificar datos y acceder a los activos de conocimiento de la organización. Asimismo, el gestor de base de datos también se ocupa de realizar consultas y hacer análisis para generar informes. Además, los sistemas de gestión de base de datos pueden entenderse como una colección de datos interrelacionados, estructurados y organizados en el ecosistema formado por dicho conjunto de programas que acceden a ellos y facilitan su gestión(Darias, 2021).

Microsoft SQL Server: Es un sistema de gestión de bases de datos relacionales (RDBMS) de Microsoft que está diseñado para el entorno empresarial desarrollado por la empresa Microsoft. Utiliza un lenguaje de programación estándar e interactivo para la obtención de información desde una base de datos y para actualizarla. Aunque SQL es a la vez un ANSI y una norma ISO, muchos productos de bases de datos soportan SQL con extensiones propietarias al lenguaje estándar.

PostgreSQL 13, o también conocido como Postgres, es un sistema de gestión de bases de datos relacionales (RDBMS) libre y de código abierto que hace énfasis en la extensibilidad y el cumplimiento de SQL. En Postgres no se requiere usar bloqueos de lectura al realizar una transacción lo cual brinda una mayor escalabilidad. También PostgreSQL tiene *Hot-Standby*. Este permite que los clientes hagan búsquedas (sólo de lectura) en los servidores mientras están en modo de recuperación o espera. Así se puede hacer tareas de mantenimiento o recuperación sin bloquear completamente el sistema(Reicek, 2017).

MySQL: es un sistema de gestión de bases de datos relacional (RDBMS). Este programa es capaz de almacenar una gran cantidad de datos y de distribuirlos para satisfacer las necesidades de cualquier tipo de organización. MySQL en sus inicios se consideraba la opción ideal para sitios web; pero actualmente incorpora muchas de las funciones necesarias para otros entornos sin perder su velocidad.

Se escoge como sistema gestor de base de datos el PostgreSQL. Permite que los fallos no afecten al resto del sistema en caso de la ocurrencia de los mismos. Además de la facilidad que brinda el PgAdmin para hacer mantenimiento de las tablas o respaldos, *Hot-Standby* permite que los

usuarios puedan acceder a las tablas en modo lectura mientras que se realizan los procesos de *backup* o mantenimiento.

Entorno de Desarrollo Integrado (IDE)

Un IDE o entorno de desarrollo integrado es una aplicación de tipo software que combina en un solo lugar todas las herramientas necesarias para realizar un proyecto de desarrollo de software. Ofrece una interfaz que permite escribir código, organizar grupos de texto y automatizar las tareas redundantes de programación. Más que un editor de código, los IDE combinan las funcionalidades de varios procesos de programación en un mismo lugar. Poseen al menos un editor, un compilador, un desbloqueador, y las funcionalidades de completación de código o de gestión de código genérico(rédac, 2022).

Visual Studio Code: Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros idiomas (como C 11 ++, C #, Java, Python, PHP) y tiempos de ejecución (como .NET y Unity). Posee depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código y fragmentos.

Apache NetBeans 11.2: NetBeans es un IDE (*Integrated Development Environment*) o entorno de desarrollo integrado, que es gratuito y de código abierto. Se debe destacar que sirve para el desarrollo de aplicaciones web, corporativas, de escritorio y móviles que utilizan plataformas como Java y HTML5, entre otras. Se trata de un editor de código fuente, junto con recursos de construcción automáticos y un depurador. Igualmente, brinda la función de autocompletado inteligente de código. Y en el caso de la plataforma de NetBeans IDE, ofrece un compilador y un intérprete. Un IDE como NetBeans permite integrar los lenguajes de programación con las plataformas correspondientes a los sistemas operativos o entornos de programación(Creana, 2022).

Se escoge Apache NetBeans como entorno de desarrollo porque entre sus funcionalidades permite escribir, depurar, compilar y ejecutar programas. Se destacan funcionalidades como el auto completamiento de código y la integración de *Frameworks* de desarrollo. Es un software multiplataforma y extensible que además posee abundante documentación referente a las clases y APIs que lo conforman.

Servidor Web

Un servidor web es una plataforma computacional muy potente que resguarda datos para ser consultados por diversos usuarios. Cuenta con un software que entrega la información solicitada por visitantes o miembros de un grupo de trabajo. La rapidez y capacidad de un servidor web serán los factores determinantes para el éxito de una página web(Coppola, 2020).

AOLServer: El servidor web AOLServer es el servidor web de código libre desarrollado por AOL (América Online, el proveedor de Internet más importante del mundo). AOL utiliza AOLServer como servidor principal de web para uno de los entornos web de mayor tráfico y uso de Internet. AOLServer es un servidor web con muchas facilidades de uso en entornos de gran escala y sitios web dinámicos. Tiene una amplia base de usuarios, debido a que está integrado con OpenACS, un sistema de gestión de contenidos muy potente, de código libre, desarrollado inicialmente por una empresa llamada ArsDigita y posteriormente liberado bajo licencia GPL(Ecured, 2018a).

Apache Tomcat: Es una aplicación de software de código abierto de las tecnologías JavaServlet y *JavaServer Pages*. Apache Tomcat se desarrolla en un entorno abierto y participativo. Es liberado y multiplataforma. Tomcat permite el monitoreo y la administración remota de las aplicaciones mediante una conexión segura(Eulises, 2020).

Se decide optar como servidor web Apache Tomcat porque está recomendado para Vaadin.

1.9 Conclusiones parciales

En este capítulo, se trataron elementos teóricos que sustentan la propuesta de solución del problema planteado, arribando a las siguientes conclusiones:

- El análisis conceptos relacionados con la gestión de información asociada a la preparación para la defensa permitió definir características y buenas prácticas que pueden ser empleadas en la propuesta de solución.
- Con la selección de la metodología, tecnologías y herramientas basadas en software libre se obtuvo una base tecnológica enfocada hacia el desarrollo de sistemas web.
- Sobre las soluciones homólogas estudiadas, se concluyó que son sistemas con un alto nivel de aceptación, pero cada una responde a necesidades particulares; aun así, fue posible identificar elementos a tener en cuenta para la propuesta de solución.

Capítulo 2

En el presente capítulo, se describen las principales características de la propuesta de solución y se identifican los requisitos funcionales y no funcionales. Se describe la arquitectura de software, los patrones de diseños y los artefactos generados para darle solución al problema planteado en la investigación.

2.1 Descripción de los procesos del negocio

Para lograr una total comprensión del negocio; a continuación, se hace una breve descripción de las características y el funcionamiento de cada uno de los principales procesos que se involucran:

- Registro del personal: el proceso de registro no es más que la recopilación y agrupación de todos los datos personales y de la defensa de cada efectivo. El registro se realiza una vez recibidos los datos actualizados de la Secretaría General, y a partir de ello, se procede a archivar el personal nuevo y actualiza los registros ya existentes.
- Asignación del personal: la asignación del personal dentro de la estructura del batallón, es tarea del personal del VDER de la facultad en coordinación con el Área de Atención Militar. El vicedecanato realiza la ubicación del personal de la facultad luego de recibir las indicaciones pertinentes para el proceso, teniendo en cuenta las necesidades existentes y en correspondencia con el resultado del análisis de los datos personales.
- Planificación de las actividades de la defensa: el Programa de Actividades del Batallón se diseña a partir del Plan de Actividades de la Defensa UCI emitido por la DSPD. En la facultad, cada actividad es el resultado de un trabajo colaborativo ente el VDER, profesores pertenecientes a la Cátedra de PPD y la máxima dirección del batallón (Plana Mayor).
- Ejecución de las actividades de la defensa: Luego de lograr un diseño de actividades acorde con las misiones y tareas del batallón, para asegurar la realización correcta y una buena participación, es responsabilidad de la Plana Mayor, a todos sus niveles, notificar y asegurar

la calidad de las mismas, así como el cumplimiento de las tareas plasmadas en la etapa de planificación.

- Evaluación de las actividades de la defensa: El último paso de este proceso es la evaluación de los resultados, para ello el VDER elabora un informe sobre la actividad donde se resume las principales observaciones y emite una evaluación correspondiente a la calidad de las actividades realizadas.

2.2 Propuesta de solución

La propuesta de solución se basa en el desarrollo del Sistema para la gestión de información asociada a la preparación para la defensa que será utilizado en la Facultad 4 de la Universidad de las Ciencias Informáticas, encargado de gestionar los procesos descritos en el epígrafe anterior; para ello debe ser capaz de:

- Crear y almacenar información personal de todos los integrantes del batallón.
- Estructurar de manera jerárquica la composición del batallón, lo que posibilita la asignación de misiones y tareas específicas.
- Disponer de información referente a Unidades Militares, así como la asignación del personal en cada una de ellas.
- Crear y permitir consultar los planes de bastiones universitarios.
- Diseñar y evaluar actividades dirigidas a cualquier estructura dentro del batallón.
- Generar reportes informativos, para favorecer la divulgación de los resultados del proceso.
- Notificar vía correo a los estudiantes de su ubicación en las unidades militares.

2.3 Definición de los actores del sistema

Un actor es una persona (o quizás otro sistema) que interactúa con el sistema para realizar un caso de uso. Provee una representación de alto nivel de los requisitos de usuario.(Fernández, 2019)

Tabla 3: Actores que intervienen en el sistema (Elaboración propia)

Actores	Justificación
Usuario	Representa el rol de los usuarios que podrán consultar información sobre actividades de la defensa.
Vicedecano de extensión y residencia (VDER)	Usuario que se encarga de gestionar toda la información referente a la defensa.
Administrador	Encargado de otorgar los roles del sistema.

2.4 Captura de Requisitos

La Ingeniería de Requisitos es un enfoque sistemático y disciplinado para la educación, documentación, validación y gestión de requisitos de software. Entre sus principales objetivos se encuentran: (1) Conocer los requisitos relevantes, logrando un consenso entre los "interesados" en estos requisitos, (2) Comprender y documentar los deseos y necesidades de los "interesados", (3) Especificar y gestionar requisitos para minimizar el riesgo de entregar un sistema que no cumpla con los deseos y necesidades de los "interesados"(Gomez, 2012).

Las técnicas aplicadas fueron la entrevista y la tormenta de ideas. Para las cuales el cliente fue incluido como parte del equipo de desarrollo, aportando y corrigiendo las ideas preliminares sobre las funcionalidades identificadas. También se realizó un procedimiento de búsqueda de documentos y archivos almacenados para determinar el funcionamiento del proceso.

2.4.1 REQUISITOS FUNCIONALES

Los requerimientos funcionales de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones (pmoinformatica.com, 2017). Como resultado de la aplicación de las técnicas de captura de información se obtuvieron 34 requisitos funcionales (RF).

A continuación, se muestra una tabla con los requisitos funcionales identificados:

Tabla 4: Requisitos funcionales identificados

No.	Nombre	Descripción	Prioridad para el cliente	Complejidad
RF1	Autenticar usuario	El usuario introduce su usuario y contraseña, si los datos son válidos el usuario queda autenticado con el nivel de privilegios asignados de acuerdo a su rol; si los datos no son válidos el sistema muestra un mensaje de error.	Alta	Alta
RF2	Incluir usuario	El sistema debe permitir incluir un usuario, solicitando los siguientes datos: <ul style="list-style-type: none"> (*) Nombre(s), (*) Apellidos, (*) Categoría, Dirección, Teléfono, (*) Sexo, Año Académico, (*) Reservista, Brigada, Grado Militar, Cargo Militar, CI. 	Alta	Media

		<ul style="list-style-type: none"> Además de los datos anteriores se debe actualizar la cantidad de usuarios registrados. 		
RF3	Modificar datos del usuario	<p>El sistema debe permitir modificar todos los datos de un usuario, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> (*) Nombre(s), (*) Apellidos, (*) Categoría, Dirección, Teléfono, (*) Sexo, Año Académico, (*) Reservista, Brigada, Grado Militar, Cargo Militar, CI. 	Media	Media
RF4	Eliminar datos de estudiantes y trabajadores que excepcionalmente se requiera.	El sistema debe permitir eliminar uno o varios usuarios.	Alta	Media
RF5	Listar usuario	<p>El sistema debe permitir listar los usuarios existentes en forma ascendente o descendente según el nombre. En el listado se deben mostrar los siguientes datos:</p> <ul style="list-style-type: none"> (*) Nombre(s), (*) Apellidos, (*) Categoría, Dirección, Teléfono, (*) Sexo, Año Académico, Reservista, Brigada, Grado Militar, Cargo Militar, CI. 		
RF6	Filtrar listado de usuarios.	El sistema debe permitir filtrar el listado de usuarios, teniendo en cuenta los siguientes datos:	Baja	Baja

		<ul style="list-style-type: none"> (*) Nombre(s), (*) Apellidos, (*) Categoría, (*) Sexo, Año Académico, Reservista, Brigada, CI. 		
RF7	Crear actividades para el aseguramiento del bastión	<p>El sistema debe permitir crear una actividad para el aseguramiento del bastión, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> (*) Actividad, (*) Fecha, (*) Hora, (*) Participantes, (*) Responsables, (*) Lugar. 	Alta	Media
RF8	Modificar actividades para el aseguramiento del bastión	<p>El sistema debe permitir modificar todos los datos de las actividades para el aseguramiento del bastión, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> (*) Actividad, (*) Fecha, (*) Hora, (*) Participantes, (*) Responsables, (*) Lugar. 	Media	Media
RF9	Eliminar actividad para el aseguramiento del bastión	<p>El sistema debe permitir eliminar una o varias actividades para el aseguramiento del bastión.</p>	Alta	Media
RF10	Crear actividades para la jornada del bastión	<p>El sistema debe permitir crear una actividad para la jornada del bastión, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> (*) Actividad, (*) Fecha, (*) Hora, (*) Participantes, (*) Responsables, (*) Orientaciones. 	Alta	Media

RF11	Modificar actividades para la jornada del bastión.	<p>El sistema debe permitir modificar todos los datos de las actividades para la jornada del bastión, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> (*) Actividad, (*) Fecha, (*) Hora, (*) Participantes, (*) Responsables, (*) Orientaciones. 	Media	Media
RF12	Eliminar actividades para la jornada del bastión.	El sistema debe permitir eliminar una o varias actividades para la jornada del bastión.	Alta	Media
RF13	Insertar estructura del batallón de la Facultad 4.	<p>El sistema debe permitir insertar una estructura del batallón, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> (*) Nombre de Estructura 	Alta	Baja
RF14	Actualizar estructura del batallón de la Facultad 4.	<p>El sistema debe permitir actualizar todos los datos de las estructuras del batallón, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> (*) Nombre de Estructura 	Media	Media
RF15	Consultar estructura del batallón de la Facultad 4	El sistema debe permitir observar la estructura del batallón.	Baja	Media

RF16	Insertar unidad militar	El sistema debe permitir insertar una unidad militar, solicitando los siguientes datos: <ul style="list-style-type: none"> (*) Nombre unidad, Descripción 	Alta	Media
RF17	Actualizar unidad militar	El sistema debe permitir actualizar todos los datos de las unidades militares, solicitando los siguientes datos: <ul style="list-style-type: none"> (*) Nombre unidad, Descripción 	Media	Baja
RF18	Listar unidad militar	El sistema debe permitir listar las unidades militares existentes en forma ascendente o descendente según el nombre. En el listado se deben mostrar los siguientes datos: <ul style="list-style-type: none"> (*) Nombre unidad, Descripción 	Alta	Media
RF19	Consultar datos de unidad militar	El sistema debe permitir ver los siguientes datos de las unidades militares: <ul style="list-style-type: none"> Nombre Unidad, Listado de Reservistas 	Baja	Baja
RF20	Eliminar datos de unidad militar	El sistema debe permitir eliminar una o varias unidades militares.	Media	Media
RF21	Consumir servicios	El sistema debe permitir consumir servicios para obtener datos de los estudiantes matriculados en la Facultad 4.	Alta	Alta

RF22	Incluir datos consumidos	El sistema debe permitir incluir automáticamente datos de los nuevos estudiantes a la base de datos del sistema, a partir de los datos previamente consumidos.	Alta	Media
RF23	Eliminar datos consumidos	El sistema debe permitir eliminar automáticamente datos de los estudiantes que han causado baja de la Universidad, a partir de los datos previamente consumidos.	Media	Media
RF24	Asignar estudiantes y trabajadores a las unidades de reservistas	El sistema debe permitir asignar estudiantes y trabajadores a unidades reservistas.	Media	Media
RF25	Eliminar reservistas de unidad militar	El sistema debe permitir eliminar reservistas asignados previamente a una unidad militar.	Baja	Baja
RF26	Actualizar la asignación de reservistas	El sistema debe permitir actualizar la asignación de estudiantes y trabajadores a las diferentes unidades reservistas.	Baja	Baja
RF27	Visualizar reservistas de la Facultad 4	El sistema debe permitir visualizar los siguientes datos de los reservistas de la facultad: <ul style="list-style-type: none"> (*) Nombre, (*) Apellidos, (*) Categoría, Cargo, Grado. 	Baja	Media

RF28	Registrar Plan de acciones del Bastión Universitario	<p>El sistema debe permitir registrar un plan de acción del bastión universitario a través de los siguientes datos:</p> <ul style="list-style-type: none"> (*) Objetivo del bastión, (*) Motivaciones del bastión, (*) Actividades para el aseguramiento del bastión, (*) Actividades para la jornada del bastión. 	Media	Media
RF29	Eliminar Plan de acciones del Bastión Universitario	El sistema debe permitir eliminar uno o varios planes de acciones del bastión universitario.	Media	Media
RF30	Buscar Plan de acciones del Bastión Universitario	<p>El sistema debe permitir buscar el plan de acción del bastión universitario, teniendo en cuenta los siguientes datos:</p> <ul style="list-style-type: none"> Nombre del plan de acción del bastión universitario. 	Media	Media
RF31	Registrar Informe resumen del Bastión Universitario	<p>El sistema debe permitir incluir un informe resumen del bastión universitario, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> (*) Asistencia, (*) Actividades de preparación, (*) Actividades de realización, Aspectos positivos, Aspectos negativos. 	Media	Baja

RF32	Eliminar Informe resumen del Bastión Universitario	El sistema debe permitir eliminar uno o varios informes del bastión universitario.	Baja	Baja
RF33	Exportar informe resumen del bastión.	El sistema debe permitir exportar en formato .pdf la información referente al informe resumen del bastión.	Baja	Baja
RF34	Exportar plan del bastión.	El sistema debe permitir exportar en formato .pdf la información referente a un plan de bastión.	Baja	Baja
RF35	Visualizar columnas.	El sistema debe permitir visualizar las columnas de las tablas que el usuario seleccione.	Baja	Baja
RF36	Consultar actividades de preparación para la defensa.	El sistema debe permitir consultar las actividades que se realizan en la preparación para la defensa.	Baja	Baja
RF37	Visitar sitios de interés.	El sistema debe permitir visitar diversos sitios de interés.	Baja	Baja
RF38	Notificar por correo	El sistema debe permitir notificar por correo para informar la ubicación de la persona en las unidades militares.	Media	Media

Con el fin de tener un mejor control de los requisitos funcionales, se procede a agruparlos por paquetes, paquetes que encapsulan funciones con un gran índice de relación entre ellas y

separados por iteraciones, atendiendo a que tan clave se consideran dentro del negocio a implementar. Estos son:

Iteración # 1 (Paquete-Personal): esta agrupación le permite al usuario consumir servicios para obtener datos personales de los estudiantes y profesores de la facultad, permitiendo en todo momento actualizar la base de datos existente. Brinda la posibilidad de crear, modificar y eliminar la información de un estudiante o de un trabajador que eventualmente se requiera, así como, hacer búsquedas avanzadas de personas y mostrar su perfil. Además, permite añadir personas a unidades militares.

Iteración #2 (Paquete-Bastión): esta agrupación permite crear la estructura del batallón, así como, la asignación de las personas a cada uno de los elementos de dicha estructura. Permite crear un plan del bastión con las actividades de aseguramiento y actividades de la jornada del mismo.

Iteración #3 (Paquete-Reportes/Consultas): esta agrupación permite realizar búsquedas avanzadas de personal, visualizar los reservistas de las unidades militares, consultar actividades de preparación para la defensa, así como, exportar reportes de planes del bastión e informe resumen de las actividades desarrolladas y notificar vía correo la unidad militar a la que pertenece cada persona.

2.4.2 REQUISITOS NO FUNCIONALES

Los requerimientos no funcionales, como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema(Sommerville, 2014).

Se trata de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema (características de usuario), sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad. En palabras más sencillas, no hablan de “lo que” hace el sistema, sino de “cómo” lo hace. A continuación, se listan y caracterizan algunos de los principales requisitos no funcionales(ISO 25010/EIC, 2011):

Eficiencia de desempeño

- Utilización de recursos: Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
- Capacidad: Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

Hardware

- Elementos físicos o materiales que constituyen una computadora o sistema informático.

Usabilidad

- Reconocibilidad de la adecuación: Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- Aprendizabilidad: Capacidad del producto que permite al usuario aprender su aplicación.

- Operabilidad: Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- Protección contra errores de usuario: Capacidad del sistema para proteger a los usuarios de hacer errores.
- Estética de la interfaz de usuario: Capacidad de la interfaz de usuario de agrandar y satisfacer la interacción con el usuario.
- Accesibilidad: Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Requisito de integridad

- Capacidad del producto para mantener su nivel de ejecución cuando se usa bajo las condiciones especificadas.

Portabilidad

- Adaptabilidad: Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
- Capacidad para ser instalado: Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.

Seguridad

- Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.

A continuación, se muestra una tabla con los principales requisitos no funcionales identificados:

Tabla 5: Requisitos no funcionales identificados

Requisitos	Descripción
Requisito de Eficiencia.	<ul style="list-style-type: none"> ✓ RnF 1 Toda funcionalidad del sistema de gestión de preparación para la defensa debe responder al usuario en menos de 3 segundos. ✓ RnF 2 La información almacenada debe ser confiable en cuanto a su veracidad e integridad desde su recopilación a través de la validación de los formularios.
Requisito de Hardware.	<p>Utilizar para pc_servidor:</p> <ul style="list-style-type: none"> ✓ RnF 3 Requerimientos mínimos: Procesador Intel(R) Core (TM) i3-4160 CPU 3.60GHz de velocidad de procesamiento, 2GB de memoria RAM y una capacidad de 40GB de disco duro. <p>Utilizar para pc_cliente:</p> <ul style="list-style-type: none"> ✓ RnF 4 Requerimientos mínimos: Procesador Pentium IV a 1.40 GHZ con 512 MB de memoria RAM.
Requisito de software.	<p>Utilizar para pc_servidor:</p> <ul style="list-style-type: none"> ✓ RnF 5 Se requiere de la instalación del servidor web Apache para poder visualizar la interfaz web. ✓ RnF 6 Se requiere de un gestor de base de datos PostgreSQL en su versión 10 o superior. <p>Utilizar para pc_cliente:</p>

	<ul style="list-style-type: none"> ✓ RnF 7 Se necesita instalado en la estación de trabajo un sistema operativo: Linux, Windows 7 o superior.
Requisito de Usabilidad.	<ul style="list-style-type: none"> ✓ RnF 8 El sistema deberá mostrar mensajes de notificación ante la ocurrencia de eventos. ✓ RnF 9 El sistema deberá ser intuitivo con el fin de establecer una navegación sencilla para el cliente. ✓ RnF 10 El sistema deberá mostrar mensajes de confirmación ante la ocurrencia de acciones críticas con el fin de proteger a los usuarios de hacer errores.
Requisito de escalabilidad.	<ul style="list-style-type: none"> ✓ RnF 11 El sistema deberá ser fácilmente escalable, con el fin de continuar perfeccionando más adelante y agregar funcionalidades.
Requisito de integridad de datos.	<ul style="list-style-type: none"> ✓ RnF 12 El sistema debe garantizar la persistencia de los datos insertados por el usuario.
Requisito de portabilidad.	<ul style="list-style-type: none"> ✓ RnF 13 El sistema debe ser capaz de ejecutarse en diferentes plataformas (Computadoras, dispositivos móviles).
Requisito de seguridad	<ul style="list-style-type: none"> ✓ RnF 14 Evitar mantener credenciales creadas por defecto, débiles o muy conocidas. ✓ RnF 15 Cerrar automáticamente la sesión de un usuario cuando ha estado inactivo durante un cierto lapso de tiempo.

2.4.3 ESPECIFICACIÓN DE LOS REQUISITOS FUNCIONALES

Historias de usuario

Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su propósito es articular cómo proporcionará una función de software valor al cliente (Atlassian, 2020).

Para mantener la organización en el proceso de desarrollo, se definió el nivel de prioridad para el negocio y el riesgo en el desarrollo de cada uno de los requisitos obtenidos en la entrevista con el cliente. Durante el diseño de la propuesta de solución se identificaron 38 historias de usuario que responden a las diferentes funcionalidades solicitadas por el cliente y presentan una descripción para que el desarrollador conozca su posterior implementación. El tiempo estimado para el desarrollo de cada uno de los requisitos se encuentra distribuido en días.

Las Historias de usuario se representan mediante tablas las cuales contienen la siguiente estructura:

Número: A cada HU se le asigna un número para facilitar su identificación por parte del equipo de desarrollo.

Nombre: Nombre descriptivo de la HU.

Programador responsable: Persona encargada de la implementación.

Prioridad en negocio: Grado de prioridad que le asigna el cliente a la HU en dependencia de sus necesidades. Los valores que puede tomar son (Alta, Media o Baja).

Riesgo en desarrollo: Grado de complejidad que le asigna el equipo de desarrollo a la HU luego de analizarla. (Alto, Medio o Bajo).

Iteración asignada: Número de la iteración en la cual será implementada la HU.

Descripción: Descripción simple sobre lo que debe hacer la funcionalidad a la que se hace referencia.

Observaciones: Condiciones que deben tenerse en cuenta para el desarrollo de la funcionalidad.

A continuación, se ilustra una historia de usuario asociada a uno de los requisitos funcionales de la primera iteración, las demás se encuentran en los anexos de trabajo:

Tabla 6: HU Incluir usuario

Historia de usuario	
Número: 2	Requisito: RF2 Incluir Usuario
Usuario:	Programador: Frank A. Valero López
Prioridad en negocio: Alta	Riesgo en desarrollo: <ul style="list-style-type: none">• Problemas de fallos eléctricos.• Planificación incorrecta.
Tiempo estimado: 2 días	Iteración asignada: 1
Descripción: 1- Objetivo: Permitir incluir nuevos usuarios al sistema. 2- Acciones para lograr el objetivo (precondiciones y datos): Para incluir un nuevo usuario hay que:	

- Tener en cuenta los siguientes datos: nombre, apellidos, teléfono, dirección, grado militar, categoría, año académico, brigada, reservista, cargo militar, sexo, ci.
- Estar autenticado en el sistema con el rol VDER.

3- Comportamientos válidos y no válidos (flujo central y alternos):

Los campos nombre, apellidos, categoría y sexo son obligatorios.

- Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres.
- Apellidos: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres.
- Categoría: Combobox con las opciones seleccionables de estudiante o profesor.
- Sexo: Combobox con las opciones seleccionables de masculino o femenino
- Reservista: Combobox con las opciones seleccionables de Si o No.

4- Flujo de la acción a realizar:

- El sistema debe permitir incluir y/o seleccionar los datos para incluir un nuevo usuario.
- Cuando se incluye y/o selecciona correctamente los datos necesarios para incluir un nuevo usuario y se selecciona la opción Guardar, se crea un nuevo elemento y el sistema muestra un mensaje de información.
- Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al usuario de realizar nuevamente la acción en cuestión.

- Si selecciona la opción Cancelar regresará a la vista previa.

Observación:

Prototipo elemental de interfaz gráfica de usuario:

2.5 Patrón Arquitectónico

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software. (Ecured, 2019)

El patrón **Modelo-Vista-Controlador (MVC)** surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo (González & Romero, 2012). A partir del uso de *frameworks* basados en el patrón MVC se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores. MVC divide las aplicaciones en tres niveles de abstracción, las cuales se describen a continuación:

En la capa del **Modelo** es donde se representa la lógica del negocio y es la encargada de acceder de forma directa a los datos actuando como intermediario con la base de datos. En esta capa actúa el ORM JPA realizando la conversión entre los objetos y las tablas de la base de datos. JPA tiene incluido el lenguaje de consulta orientado a objetos JPQL (*Java Persistence Query Language*,

Lenguaje de Consulta de Persistencia en Java) que permite obtener más de un objeto por consulta(Alicante, 2018).

La capa de la **Vista** o de presentación es la encargada de mostrar la información al usuario de forma gráfica y legible. El *framework* Vaadin estará presente en esta capa ya que está diseñado para trabajar en la vista.

La capa del **Controlador** es la intermediaria entre la vista y el modelo. Controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que esta lo presente al usuario, de una manera legible(Alicante, 2018). En esta capa estarán las clases Java de la aplicación.

A continuación, se muestra una figura que describe el funcionamiento de la arquitectura de la propuesta de solución.

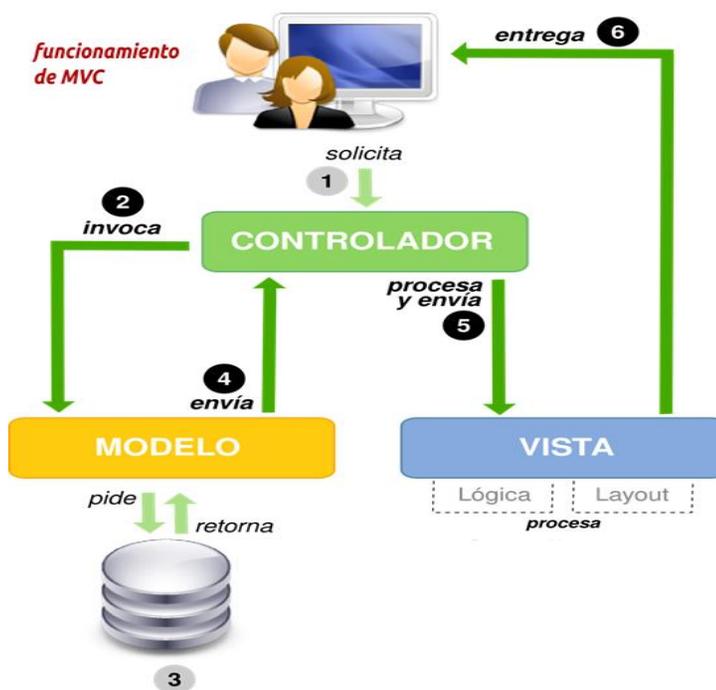


Ilustración 3: Funcionamiento del patrón arquitectónico MVC

En resumen, la vista muestra el modelo al usuario, el usuario a su vez, activa eventos para que la vista se envíe al controlador, quien recibe y manipula el modelo dependiendo de la solicitud. El modelo realiza las operaciones necesarias para modificar la información y notificar al controlador para que finalmente éste decida mostrar una nueva vista al usuario, la cual posee el modelo ya actualizado. Los componentes actúan de forma independiente con actividades definidas, encapsularlas permite a los desarrolladores una mayor reutilización de estos, mejor mantenimiento y por supuesto, asegura la construcción de una aplicación escalable sin afectar los componentes desarrollados anteriormente.

2.6 Patrones de Diseño

Los patrones de diseño son soluciones habituales a problemas que ocurren con frecuencia en el diseño de software. Son como planos prefabricados que se pueden personalizar para resolver un problema de diseño recurrente en tu código(Cosmin, 2016).

PATRONES GRASP

Los patrones GRASP (*General Responsibility Assignment Software Patterns*), son patrones generales para la asignación de responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, y sirven para asignar correctamente las responsabilidades en el diseño de la Programación Orientada a Objetos (POO)(Quintero, 2020). A continuación, se detalla el uso de estos patrones en la solución.

Alta cohesión: es la medida en la que un componente o clase realiza únicamente la tarea para la cual fue diseñada (Una clase debe de hacer lo que respecta a su entidad, y no hacer acciones que involucren a otra clase o entidad). Este patrón se aplica en las clases controladoras `EstudiantesView.java`, `ResumenView.java`, entre otras, las cuales se encargan de la implementación de las funcionalidades que le corresponden haciendo uso solo de sus entidades asociadas (estudiantes, resumen).

Creador: El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. El empleo de este patrón se ve reflejado en la clase `BatallonService`.

Experto: es el principio básico de asignación de responsabilidades. Indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. En la presente solución se utiliza el mencionado patrón dentro de la capa de acceso a datos, se evidencia en las clases entidades, las cuales son expertas en su propia información, tales como `Estudiante` y `Unidades Militares`.

Bajo Acoplamiento: es la medida en que un componente o clase dependen de otro, generando cambios externos o alterando la funcionalidad del mismo (Cuando se modifican los atributos de una clase, se tienen que modificar los atributos de otra clase)(Solve, 2022). Se aplica en las clases controladoras `AseguramientoView.java`, `PlanBastionView.java`, entre otras.

La siguiente ilustración muestra como se ven reflejados estos patrones en el código:

```

@Service
public class BatallonService {

    private final BatallonRepository repository;

    public BatallonService(@Autowired BatallonRepository repository) {
        this.repository = repository;
    }

    public List<Batallon> findAll() {
        return repository.findAll();
    }

    public Batallon findById(Integer id) {
        return repository.findById(id).get();
    }

    public Batallon save(Batallon batallon) {
        return repository.save(batallon);
    }
}

```

```

@Data
@NoArgsConstructor
@AllArgsConstructor
@EqualsAndHashCode(onlyExplicitlyIncluded = true)
@Entity
public class UnidadesMilitares extends AbstractEntity {

    @EqualsAndHashCode.Include

    @Column
    private String nombreUnidad;

    @Column
    private String descripcionUnidad;
}

```

Creador **Experto**

Ilustración 4: Ejemplo de los patrones en el código del sistema

Patrones GOF

Patrones **GOF** (*Gang of four*, por sus siglas en inglés), son clasificados según el propósito para el cual han sido definidos. Entre los patrones usados se encuentran los siguientes:

Instancia única (Singleton): Garantiza que exista una única instancia de una clase. Si los módulos y los temas se piensan como objetos, este patrón está presente, debido a que en general estos objetos no encapsulan datos, lo que separa a un módulo de otro es el conjunto de funciones que contiene. Esto garantiza la presencia de una única instancia y el mecanismo de acceso global a ella (Andalucía, 2020).

2.7 Modelo del diseño

El modelo de diseño está adaptado para modelar el entorno de implementación real, y sirve como una abstracción del código fuente. Se utiliza para concebir y para documentar el diseño del sistema

de software. Es un producto de trabajo integral y compuesto que abarca clases del diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos(Sommerville, 2014).

Diagrama de clases

Un diagrama de clases es una representación de las clases que involucran al sistema y las relaciones entre ellas que pueden ser de asociación, de herencia, de uso y de agregación. Una clase es una definición de un conjunto de entidades u objetos que comparten los mismos atributos, operaciones y relaciones(Volodina, 2019). La siguiente figura muestra el Diagrama de clases del diseño con estereotipos web que responde a un conjunto de funcionales del sistema a implementar.

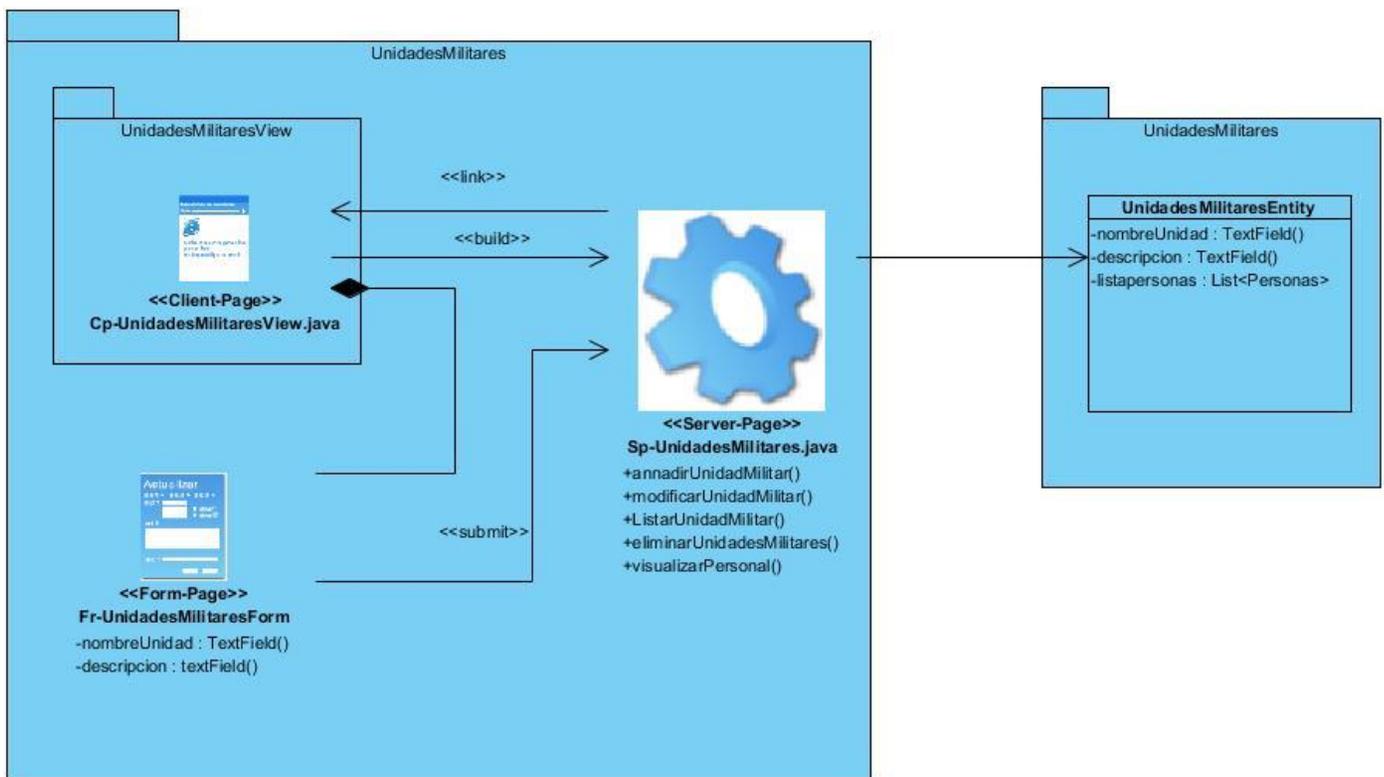


Ilustración 5: Diagrama de Clases (Elaboración propia)

La figura antes ilustrada representa el diagrama de clases para un conjunto de objetos que comparten los mismo atributos y operaciones referente a las unidades militares. En ella se puede observar la forma en la que interactúan dichos objetos para el correcto funcionamiento de sus operaciones.

Modelo de datos

Un diagrama entidad-relación, también conocido como modelo entidad relación o ERD, es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Los diagramas ER se usan a menudo para diseñar o depurar bases de datos relacionales en los campos de ingeniería de software.

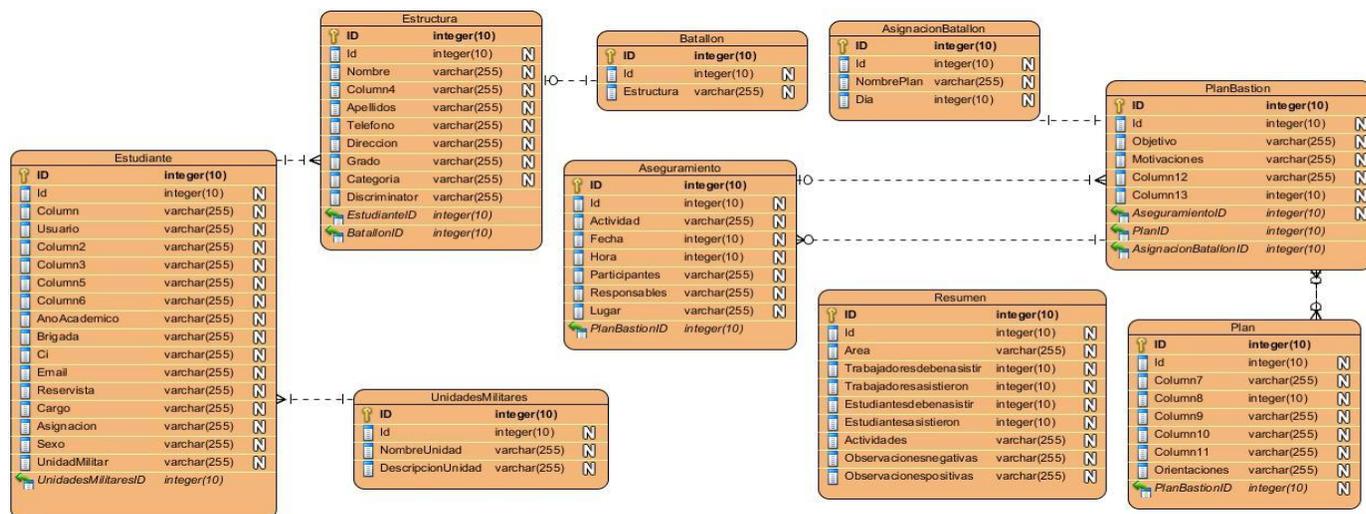


Ilustración 6: Diagrama Entidad-Relación

En el anterior diagrama se muestran las principales entidades del negocio con los pertinentes atributos asociados a las mismas. Este diagrama ayuda a comprender mejor las entidades presentes, así como, la relación que se establece entre ellas.

2.8 Modelo de implementación

El modelo de implementación representa la composición física de la implementación. Define las principales unidades de integración alrededor de las cuales se organizan los equipos, así como las unidades que se pueden versionar, desplegar o reemplazar separadamente. También puede incluir código fuente de nivel bajo y archivos derivados, y sus relaciones con el modelo de diseño (Guadarrama, 2022).

Diagrama de componentes

El diagrama de componentes tiene como objetivo describir la relación que existe entre los distintos componentes del sistema. Está directamente vinculado con el diseño del sistema, permitiendo modelar las relaciones e interfaces que existen entre los componentes. Está orientado a la implementación del sistema. Está compuesto por componentes, interfaces y sus relaciones. (Dirección de Calidad UCI, 2020) La siguiente figura refleja los componentes a utilizar en la implementación de uno de los procesos que maneja el sistema de gestión de preparación para la defensa.

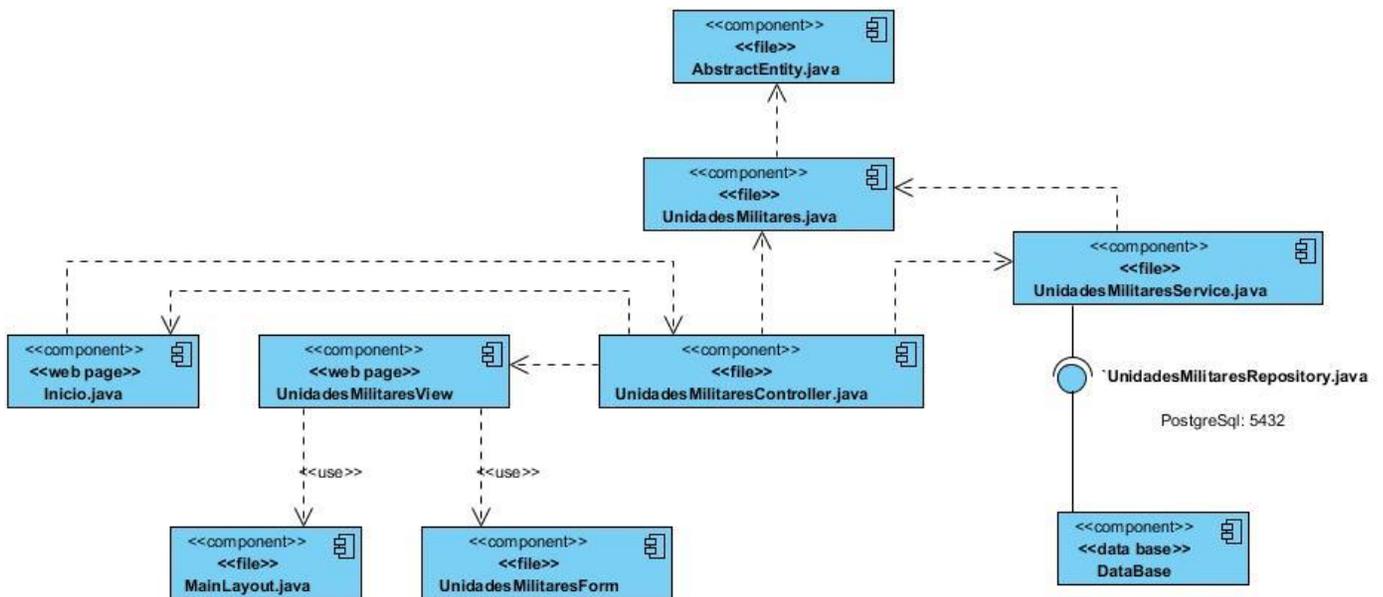


Ilustración 7: Diagrama de componentes (Elaboración propia)

La figura ilustrada anteriormente refleja el diagrama de componentes para uno de los procesos que se manejan en el sistema. En ella se pueden observar las relaciones que existen entre dichos componentes y las interfaces a través de sus relaciones lo cual orienta a la implementación del sistema.

2.9 Diagrama de despliegue

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Describen la topología del sistema la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP (Ecured, 2018b).

A continuación, se muestra el ambiente donde debe ser desplegada la propuesta solución:

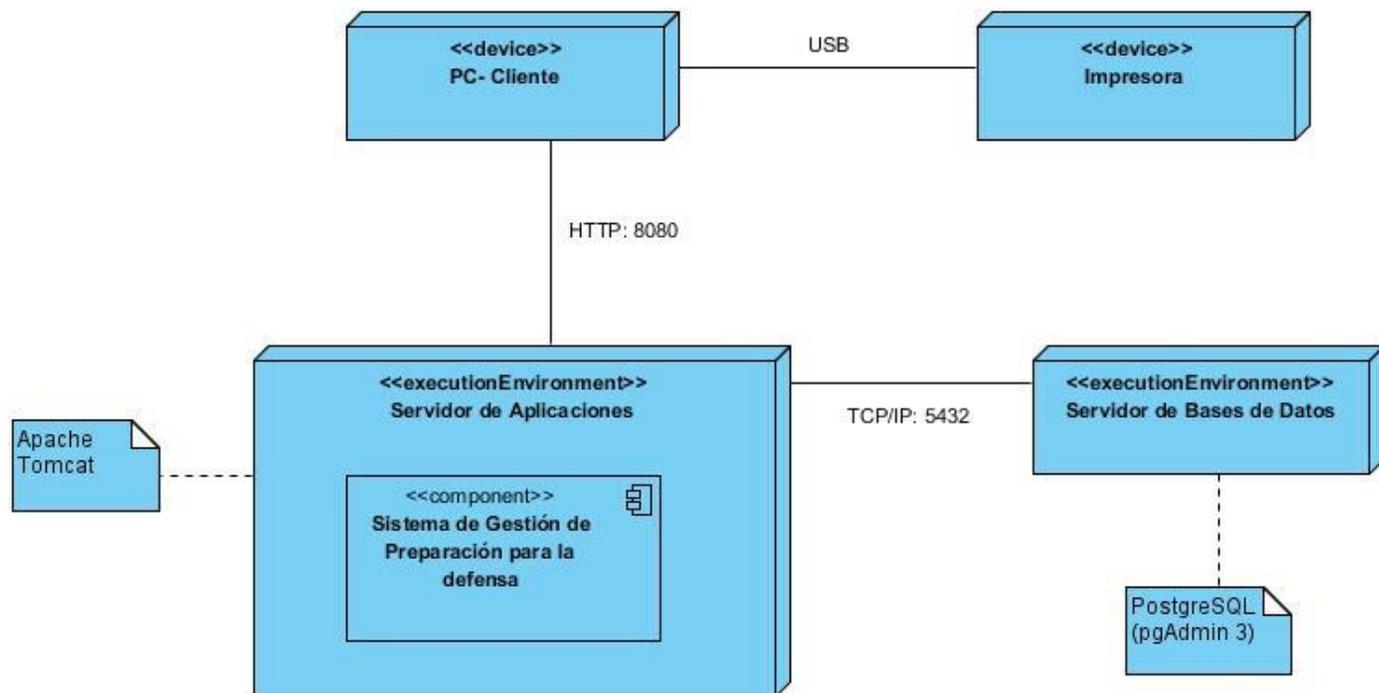


Ilustración 8: Diagrama de despliegue (Elaboración propia)

Descripción de los nodos

- **Servidor Web:** representa una estación donde estará montado el servidor Apache Tomcat sobre el cual se estará ejecutando la aplicación.
- **Cliente:** representa el dispositivo desde el cual el usuario podrá acceder a la aplicación.
- **Impresora:** representa el dispositivo que imprimirá los reportes ofrecidos por la aplicación.
- **Servidor de Bases de Datos:** representa el servidor donde estará el sistema gestor de base de datos PostgreSQL que dará respuesta a las peticiones hechas por la aplicación.

Protocolos

- **HTTP (Protocolo de transferencia de hipertexto):** establece un esquema de comunicación cliente/servidor. El cliente es el navegador web que realiza las peticiones a las que el servidor se encarga de dar respuesta.
- **USB (Bus Universal en Serie):** permite conectar periféricos para atender peticiones.
- **TCP/IP (Protocolo de control de transferencia):** se utiliza en la comunicación entre el servidor y la base de datos para realizar operaciones sobre la información de las tablas.

Con la elaboración del diagrama de despliegue se pueden visualizar qué elementos de software se implementan mediante qué elementos de hardware. Además, se conocen los enlaces de comunicación entre estos componentes.

2.10 Conclusiones parciales

Una vez transitado por las distintas disciplinas de la metodología AUP se concluye que:

1. Los requisitos funcionales y no funcionales que fueron identificados y encapsulados en historias de usuario, atendiendo a lo normado para el desarrollo de la disciplina de Requisitos

de la metodología de desarrollo AUP-UCI propiciaron el acuerdo entre desarrolladores y clientes sobre las características del sistema a desarrollar.

2. Los artefactos correspondientes a la disciplina de Análisis y Diseño, construidos a partir de los requisitos definidos, representaron las características del sistema con elementos que constituyen base directa para la posterior etapa de implementación.

Capítulo 3

En este capítulo se abordarán las pruebas realizadas al software siguiendo las buenas prácticas de la metodología AUP en su versión UCI para garantizar el correcto funcionamiento la calidad del sistema. Una vez concluida la fase de pruebas se podrá definir con exactitud, en qué grado de funcionamiento y rendimiento se encuentra el sistema, compararlo con las exigencias del cliente y determinar si se cumplieron sus expectativas.

3.1 Fase de Pruebas

Las pruebas de software forman parte de la última fase que propone la metodología AUP versión UCI en su etapa de cierre. Las pruebas, por lo general involucran operaciones del sistema evaluando los resultados bajo condiciones controladas.

El proceso de pruebas permite evaluar y verificar la calidad del producto, controlando su correcto funcionamiento según los requisitos definidos por el cliente, evitando de esta manera el número de errores no deseados, reduciendo los costos para el desarrollo de la aplicación y garantizando una mejora en el rendimiento del sistema.

3.2 Estrategia de pruebas

Una estrategia de prueba del software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de los casos de prueba, la ejecución de las pruebas y la agrupación y evaluación de los datos resultantes.(Pressman, 2005), (Beizer, 1984) Las estrategias de prueba, de forma general, pueden incluir varios niveles de pruebas, que permiten garantizar un resultado final satisfactorio. Estos niveles de pruebas son(Pressman, 2005):

- Pruebas unitarias: consiste en la verificación de unidades del software (componente de software o módulo) de forma aislada, es decir, probar el correcto funcionamiento de una unidad de código.

- Pruebas de integración: Es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados mediante la prueba de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.
- Pruebas de sistema: se realizan una vez que se han probado los componentes y la integración de los mismos, con el objetivo de estudiar los requisitos funcionales y no funcionales del sistema y las características de calidad.
- Pruebas de aceptación: son responsabilidad del cliente y pueden ser la única parte de las pruebas en donde estén involucrados. Se llevan a cabo antes de que el programa se ponga en funcionamiento real y tienen que satisfacer las expectativas del cliente.

Para el desarrollo de pruebas en los diferentes niveles, son utilizados métodos de pruebas de software, que incluyen la identificación, medición y evaluación de una o más cualidades, características o propiedades del producto de software en cuestión.

3.2.1 PRUEBAS UNITARIAS

Con las pruebas unitarias se verifican las condiciones límites para asegurar que la unidad de software funciona correctamente en las fronteras establecidas como restricciones de procesamiento. Se deben diseñar casos de prueba para detectar errores debido a cálculos y comparaciones incorrectas o flujos de control inapropiados (Pressman, 2005) y (KeepCoding, 2022).

En este nivel, se seleccionó la utilización del método de caja blanca. Este método examina la parte interna del programa. Mediante el método de prueba de caja blanca, pueden obtenerse casos de prueba que garanticen que se (Pressman, 2005):

- ejerciten por lo menos una vez todos los caminos independientes de cada unidad de software,
- ejerciten todas las decisiones lógicas en sus vertientes verdadera o falsa,
- ejecuten todos los bucles en sus límites y con sus límites operacionales y
- ejerciten las estructuras internas de datos para asegurar su validez.

Las características de este método de prueba implican que tengan una alta correspondencia con el nivel de pruebas unitarias, debido a que permiten verificar unidades específicas de pequeño tamaño. Se basa en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa y contar con la disponibilidad del código fuente.

3.2.2 PRUEBAS DE SISTEMA

Las pruebas de sistema tienen como propósito primordial ejercitar profundamente el sistema basado en computadora. Trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas(Pressman, 2005).

La realización de pruebas a los requisitos funcionales en este nivel se basó en la utilización del método de caja negra. Este método se enfoca en probar el sistema sin tomar en cuenta la estructura interna del mismo. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar. Su objetivo es validar que el comportamiento de las salidas del sistema sean las esperadas. Para ello, se obtiene un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa de software. Si las salidas no son las esperadas, entonces la prueba ha detectado un problema en el funcionamiento del software(Pressman, 2005a).

La prueba de caja negra no es una alternativa a la prueba de caja blanca. Se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores de las siguientes categorías(Pressman, 2005b):

- funciones incorrectas o ausentes,
- errores de interfaz,
- errores en estructuras de datos o en accesos a bases de datos externas,
- errores de rendimiento y
- errores de inicialización y de terminación.

Las pruebas no funcionales a realizar se seleccionaron, para este nivel, teniendo en cuenta las características no funcionales y la complejidad, en cuanto a desarrollo, que implica cada una de ellas. Son ellas:

Prueba de seguridad: intenta verificar que los mecanismos de protección incorporados en el sistema impedirán los accesos impropios(Pressman, 2005). Esta prueba permitirá comprobar que al sistema de preparación para la defensa solamente accederán los usuarios autorizados.

Prueba de volumen: hace referencia a grandes cantidades de datos para determinar los límites en que se causa que el sistema falle.

Prueba de carga: se ejecuta con el objetivo de determinar y validar la respuesta del software cuando está sometida a una carga de un cierto número de usuarios o de peticiones. Este resultado se compara con un volumen esperado. Ejecutar este tipo de pruebas al producto de software resultante de esta investigación resultará relevante para la prevención de posibles fallos, debido a que al consumir los servicios UCI para obtener los datos de las personas de la Facultad 4 implica una gran concentración de datos.

3.3 Pruebas de aceptación

Se considera que las pruebas de aceptación arrojan un resultado satisfactorio cuando el software funciona de acuerdo con las expectativas razonables del cliente. Las pruebas de aceptación a menudo se realizan en dos etapas(Pressman, 2005):

- Prueba alfa: se lleva a cabo por el cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado.
- Prueba beta: se realizan con posterioridad a las pruebas alfa, y se desarrollan en el entorno del cliente. En este caso, el cliente se queda a solas con el producto y trata de encontrarle fallos de los que informa al desarrollador.

Dentro de la estrategia de pruebas que se planifica, la validación del producto de software desarrollado, por parte del cliente, se incluye con la variante de pruebas de aceptación alfa. Es una variante viable porque desde etapas tempranas se concibió que la aceptación del módulo fuera ofrecida por la persona que ha representado a la entidad cliente durante todo el proceso de

desarrollo, por tanto, resulta cómodo su traslado hacia el lugar donde se ha desarrollado el producto. Además, otro aspecto favorable es que conoce a los desarrolladores y el entorno donde se realizarán las pruebas; lo que incide en la idoneidad del ambiente de pruebas.

3.3 Aplicación de pruebas de aceptación

La partición de equivalencia es una técnica de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Básicamente, este método intenta dividir el dominio de entrada de un programa en un número finito de clases de equivalencia (datos de entrada), de tal modo que se pueda asumir razonablemente que una prueba realizada con un valor representativo de cada clase es equivalente a una prueba realizada con cualquier otro valor de dicha clase. Esto quiere decir que, si el caso de prueba correspondiente a una clase de equivalencia detecta un error, el resto de los casos de prueba de dicha clase de equivalencia deben detectar el mismo error, y viceversa; si un caso de prueba no ha detectado ningún error, es de esperar que ninguno de los casos de prueba correspondientes a la misma clase de equivalencia encuentre ningún error (Pressman, 2005).

Se realizaron varios casos de prueba usando la técnica mencionada anteriormente, a continuación, se muestra un ejemplo de los mismos y los demás casos de pruebas se presentan en los anexos:

Tabla 7: Clases de equivalencia

Identificador	Entrada(clase)	Clases válidas	Clases inválidas
Actividad	Actividad	Actividad = (a-z)	Actividad = (0-9), caracteres especiales.
Fecha	Fecha	Fecha = (D/M/A)	Fecha = cualquier otro formato de fecha

Hora	Hora	Hora = (h:m)	Hora = cualquier otro formato de hora
Participantes	Participantes	Participantes = (a-z)	Participantes = (0-9), caracteres especiales
Responsables	Responsables	Responsables = (a-z)	Responsables = (0-9), caracteres especiales
Lugar	Lugar	Lugar = (a-z), (0-9), caracteres especiales	Lugar = ninguna

Descripción general: Permitir incluir nuevas actividades para el aseguramiento del bastión.

Condiciones de ejecución: Para incluir una nueva actividad para el aseguramiento del bastión hay que: Estar autenticado en el sistema con el rol VDER.

SC 1 Incluir Actividad para el aseguramiento del bastión.

Tabla 8: Diseño de caso de prueba incluir actividad para el aseguramiento del bastión

Esce- nario	Descripción	Ac- tivi- dad	Fe- cha	Hora	Par- tici- pan- tes	Res- pon- sables	Lu- gar	Respuesta del sistema	Flujo cen- tral
EC 1.1 Opción Incluir activi- dad de asegu- ra- miento para el bas- tión.	Selecciona la opción de incluir una nueva actividad para el aseguramiento para el bastión.	V	V	V	V	V	V	Brinda la posibilidad de introducir o seleccionar de manera obligatoria los siguientes datos de la actividad de aseguramiento: - Actividad – Fecha – Hora - Participantes – Responsables - Lugar. Permite: - Añadir los datos. - Cancelar la operación en cualquier momento.	Plan Bas- tión/In- cluir.
EC 1.2 Opción de	Introduce y/o selecciona los datos de	V	V	V	V	V	V	Valida los datos. Crea una actividad para el aseguramiento del bastión.	Plan Bas- tión/In- cluir/Añadir

Añadir los datos.	la actividad de aseguramiento y selecciona la opción guardar los datos.							Muestra el listado de actividades para el aseguramiento del bastión y un mensaje de información: "Actividad añadida".	
EC 1.3 Opción de cancelar.	Selecciona la opción de Cancelar.							<i>Elimina los datos creados.</i> Regresa al listado de actividades para el aseguramiento del bastión.	Plan Bastión/Incluir/Cancelar
EC 1.4 Datos incompletos	Existen datos incompletos.	I	V	V	V	V	V	Deshabilita la opción añadir. Muestra un indicador sobre los campos vacíos y un mensaje: "El campo no puede estar vacío". <u>Regresa al EC 1.1.</u>	Plan Bastión/Incluir/Añadir
		V	I	V	V	I	V		
		V	V	I	V	V	V		
EC 1.5 Datos incorrectos	Existen datos incorrectos.	V	V	V	I	V	V	Muestra un mensaje de información. Muestra un indicador sobre los campos incorrectos. <u>Regresa al EC 1.1.</u>	Plan Bastión/Incluir/Añadir
		V	V	V	V	I	V		

Se realizaron pruebas funcionales al sistema para comprobar la correcta implementación de los requisitos funcionalidades del mismo. Para asegurar que las no conformidades detectadas en cada iteración fueran corregidas, se aplicaron pruebas de regresión antes de pasar a la próxima iteración. Se llevaron a cabo tres iteraciones de pruebas funcionales durante las pruebas del sistema. En cada una de las iteraciones, se detectaron diferentes no conformidades clasificadas en bajas, medias y altas. Las bajas y medias, se centraron en errores ortográficos, como omisiones de tildes y cambio de mayúscula por minúscula, formatos incorrectos y validaciones de campos. Las altas consistieron en errores de funcionalidad.

En la primera iteración de pruebas funcionales se identificaron 9 no conformidades realizadas a un total de 18 requisitos funcionales y clasificadas en bajas, medias y altas. En la segunda iteración se realizaron pruebas de regresión para verificar que las no conformidades de la iteración anterior se habían resuelto, además se identificaron 6 no conformidades de clasificación baja y media. En la tercera y última iteración se comprobó que todas las no conformidades detectadas fueron

resueltas y surgen 4 nuevas no conformidades. La siguiente tabla muestra el total de no conformidades según los tipos de error detectados.

Tabla 9: Cantidad de no conformidades según el tipo de error y riesgo detectado

Tipo de error	Primera iteración	Segunda iteración	Tercera iteración	Cantidad de no conformidades	Riesgo
Ortográficos	2	2	3	7	Bajo
Formato	2	1	0	3	Bajo
Validaciones	2	2	1	5	Medio
Funcionales	3	1	0	4	Alto
	Total = 9	Total = 6	Total = 4	Total = 19	

La ejecución de la totalidad de las pruebas de caja negra planificadas permitió comprobar el comportamiento satisfactorio del sistema de gestión desarrollado para cada uno de los casos de prueba diseñados. La siguiente figura visualiza la cantidad de no conformidades clasificadas y distribuidas por iteraciones durante el proceso de pruebas en las diferentes iteraciones del desarrollo correspondiente.

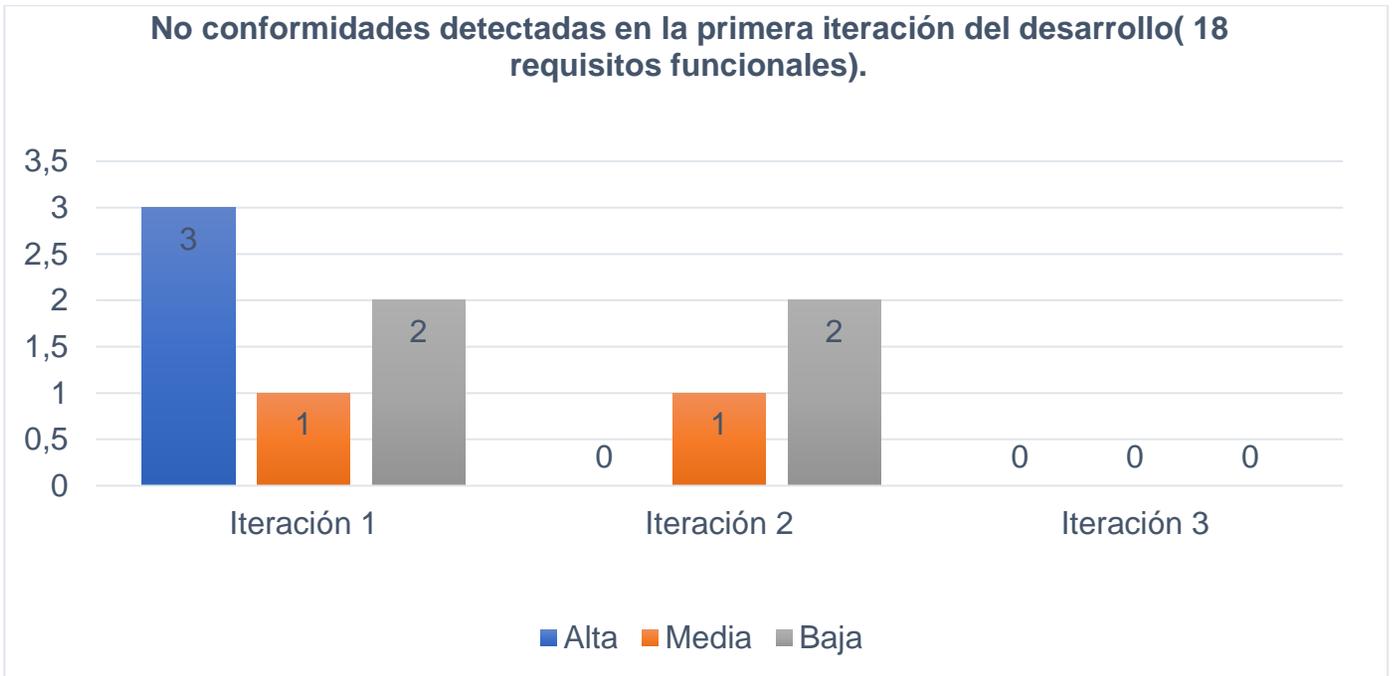


Ilustración 9: Clasificación de las no conformidades para una primera iteración de desarrollo (18 requisitos funcionales.)

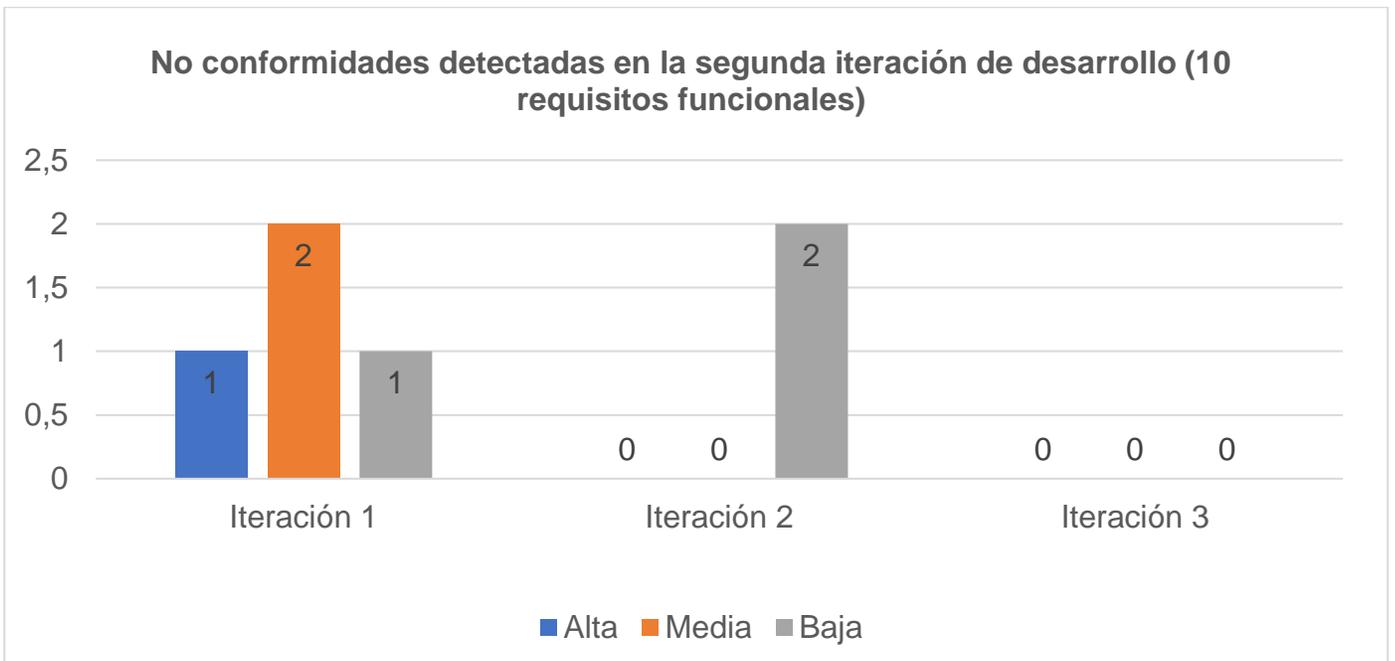


Ilustración 10: Clasificación de las no conformidades para la segunda iteración de desarrollo (10 requisitos funcionales.)

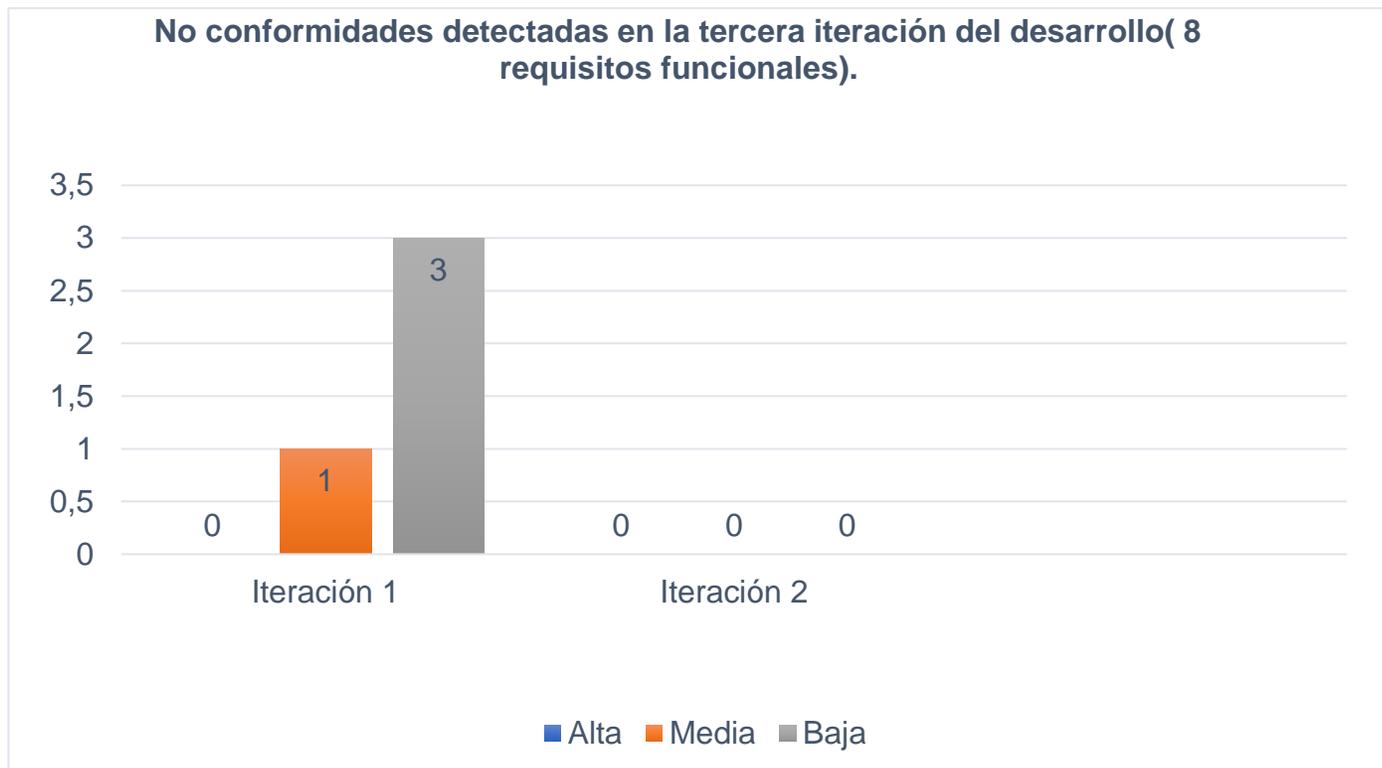


Ilustración 11: Clasificación de las no conformidades para la tercera iteración de desarrollo (8 requisitos funcionales.)

Las pruebas de aceptación fueron diseñadas como un mecanismo útil para que el cliente del sistema, quien además representa un usuario final potencial, gane confianza en la solución de software desarrollada. Este fue también el objetivo que se estableció para la etapa de ejecución de estas pruebas. Fueron probados 38 requisitos funcionales, que representan un 100% del total definido, incluidos en la planificación para este nivel de pruebas.

3.4 Conclusiones parciales

La definición y ejecución de una estrategia de pruebas permitieron concluir que:

1. La ejecución de pruebas de software en diferentes niveles permitió comprobar, con la profundidad necesaria, la satisfactoria implementación del sistema de gestión de preparación para la defensa.
2. La aplicación del método de caja blanca, empleando la técnica de camino básico, facilitó que se identificaran errores existentes y que estos fueran corregidos tempranamente.
3. La ejecución de las pruebas de caja negra, a través del método de partición de equivalencia con el diseño de los casos de prueba correspondientes, posibilitó la identificación y eliminación de no conformidades que afectaban moderadamente el funcionamiento de la solución de software construida.

CONCLUSIONES GENERALES

Una vez finalizados la investigación y el desarrollo asociados al sistema, se enuncian las siguientes conclusiones:

1. Los principales artefactos que propone la metodología AUP-UCI en su escenario 4 fueron desarrollados y garantizaron la documentación e implementación correctas y estandarizadas.
2. El sistema para la gestión de información asociada a la preparación para la defensa en la Facultad 4 de la Universidad de las Ciencias Informáticas fue implementado sobre la base de buenas prácticas para procesos de desarrollo de software.
3. La ejecución de pruebas de software definidas permitió la comprobación de la correspondencia de las funcionalidades incluidas en el sistema desarrollado con las necesidades e intereses del cliente, demostrando la obtención de un producto de alta calidad.

RECOMENDACIONES

Se recomienda ampliar hasta un nivel UCI el campo de acción donde se empleará la solución propuesta.

Referencias bibliográficas

- Alicante, S. de I. U. de. (2018, noviembre 20). Modelo vista controlador (MVC). Recuperado 26 de octubre de 2022, de <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- Andalucía. (2020). Singleton | Marco de Desarrollo de la Junta de Andalucía. Recuperado 1 de diciembre de 2022, de <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/202>
- Atlassian. (2020). Historias de usuario | Ejemplos y plantilla | Atlassian. Recuperado 30 de octubre de 2022, de <https://www.atlassian.com/es/agile/project-management/user-stories>
- Beizer, B. (1984). *Software System Testing and Quality Assurance*. Van Nostrand Reinhold.
- Bunch, L. M. (2011, noviembre 9). Ministerio de las Fuerzas Armadas Revolucionarias (Cuba)—EcuRed. Recuperado 23 de octubre de 2022, de [https://www.ecured.cu/Ministerio_de_las_Fuerzas_Armadas_Revolucionarias_\(Cuba\)](https://www.ecured.cu/Ministerio_de_las_Fuerzas_Armadas_Revolucionarias_(Cuba))
- Coppola, M. (2020, marzo 10). Qué es un servidor web, para qué sirve, cómo funciona y ejemplos. Recuperado 25 de octubre de 2022, de <https://blog.hubspot.es/website/que-es-servidor-web>
- Cosmin, P. I. (2016). Patrones de Diseño. *MoleQla: revista de Ciencias de la Universidad Pablo de Olavide*, (23), 36.
- Creana. (2022).  ¿Qué es Netbeans y para qué sirve? [2022]. Recuperado 24 de octubre de 2022, de [Https://www.crehana.com](https://www.crehana.com) website: <https://www.crehana.com/blog/transformacion-digital/que-es-netbeans/>

Darias. (2021, septiembre 9). Gestor de Base de datos: Qué es, Funcionalidades y Ejemplos.

Recuperado 24 de octubre de 2022, de Intelequia website:

<https://intelequia.com/blog/post/2949/gestor-de-base-de-datos-qué-es-funcionalidades-y-ejemplos>

Dirección de Calidad UCI. (2020). *Modelo de implementación—Expediente de Proyecto 5.0.*

Recuperado de <http://mejoras.prod.uci.cu>

Ecured. (2015). Metodología de Desarrollo de Software Variación de AUP para la UCI - EcuRed.

Recuperado 24 de octubre de 2022, de

https://www.ecured.cu/Metodolog%C3%ADa_de_Desarrollo_de_Software_Variaci%C3%B3n_de_AUP_para_la_UCI

Ecured. (2018a). AOLServer—EcuRed. Recuperado 25 de octubre de 2022, de

<https://www.ecured.cu/AOLServer>

Ecured. (2018b). Diagrama de despliegue—EcuRed. Recuperado 31 de octubre de 2022, de

https://www.ecured.cu/Diagrama_de_despliegue

Ecured. (2019, julio 15). Arquitectura de software—EcuRed. Recuperado 26 de octubre de 2022,

de https://www.ecured.cu/Arquitectura_de_software

Equipo Editorial. (2021). Lenguaje de Programación—Concepto, tipos y ejemplos. Recuperado 24

de octubre de 2022, de Concepto website: <https://concepto.de/lenguaje-de-programacion/>

Esan. (2021, noviembre 8). ¿En qué se diferencian las metodologías ágiles y tradicionales? |

Conexión ESAN. Recuperado 2 de noviembre de 2022, de

<https://www.esan.edu.pe/conexion-esan/en-que-se-diferencian-las-metodologias-agiles-y-tradicionales>

Eulises. (2020, marzo 26). ¿Qué es Apache Tomcat? Recuperado 25 de octubre de 2022, de Blog HostDime Argentina, servidores dedicados website:

<https://www.hostdime.com.ar/blog/que-es-apache-tomcat/>

Fernández. (2019). Especificación de Requisitos del Sistema | Marco de Desarrollo de la Junta de Andalucía. Recuperado 27 de octubre de 2022, de

<https://www.juntadeandalucia.es/servicios/madeja/sites/default/files/historico/1.3.0/contenido-recurso-407.html>

Fernández, F. (2017, abril 16). ¿QUE ES GESTIÓN DE LA INFORMACIÓN? Recuperado 23 de octubre de 2022, de <https://instituciones.sld.cu/toximed/2017/04/16/que-es-gestion-de-la-informacion/>

GlosarioIT. (2022). Herramienta CASE - GlosarioIT: Glosario Informático. Recuperado 2 de noviembre de 2022, de https://www.glosarioit.com/Herramienta_CASE

Gomez, V. H. O. (2012). *Ld-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF*.

Recuperado de

https://www.academia.edu/34479503/Ld_Ingenieria_de_software_enfoque_practico_7ed_Pressman_PDF

González, Y. D., & Romero, Y. F. (2012). Patrón Modelo-Vista-Controlador. *Telemática*, 11(1), 47-57.

- Guadarrama, J. (2022). MODELO DE IMPLEMENTACIÓN. Recuperado 3 de noviembre de 2022, de Prezi.com website: <https://prezi.com/ijpanrlnvhyj/modelo-de-implementacion/>
- ISO 25010/EIC. (2011). ISO 25010. Recuperado 13 de noviembre de 2022, de <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
- KeepCoding, R. (2021, septiembre 15). 5 lenguajes de programación más usados en 2022. Recuperado 25 de octubre de 2022, de <https://keepcoding.io/blog/5-lenguajes-de-programacion-mas-usados-2022/>
- KeepCoding, R. (2022, agosto 1). ¿Qué son las pruebas unitarias de software? Recuperado 31 de octubre de 2022, de <https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/>
- Kenjo. (2022). Sistemas de Información de Recursos Humanos (SIRH). Recuperado 25 de octubre de 2022, de <https://blog.kenjo.io/es/que-es-un-sistema-de-informacion-de-recursos-humanos>
- Kyocera. (2021). Los 6 principales tipos de sistemas de información | Kyocera. Recuperado 24 de octubre de 2022, de <https://www.kyoceradocumentsolutions.es/es/smarter-workspaces/business-challenges/the-cloud/los-6-principales-tipos-sistemas-informacion.html>
- López, N. (2016). [PDF] Estándares de codificación Java—Free Download PDF. Recuperado 3 de noviembre de 2022, de <https://silo.tips/download/estandares-de-codificacion-java>

- Martínez, E. S. (2014). Propuesta de procedimiento para realizar pruebas de caja blanca a las aplicaciones que se desarrollan en lenguaje Python. *3C TIC. Cuadernos de desarrollo aplicados a las TIC*, 3(2), 89-114. <https://doi.org/10.17993/3ctic.2014.32.89> - 114
- Michael «Monty». (2022). Vaadin. En *Wikipedia, la enciclopedia libre*. Recuperado de <https://es.wikipedia.org/w/index.php?title=Vaadin&oldid=145465148>
- MINED. (2021, octubre 15). Preparación para la Defensa Civil—Ministerio de Educación de la República de Cuba. Recuperado 23 de octubre de 2022, de <https://www.mined.gob.cu/defensa-civil/preparacion-para-la-defensa-civil/>
- Minfar. (2022, noviembre 1). Defensa y Seguridad Nacional. Recuperado 2 de noviembre de 2022, de Sitio Oficial del Ministerio de las Fuerzas Armadas Revolucionarias website: <https://www.minfar.gob.cu/defensa-nacional>
- Muradas, Y. (2018, junio 5). Qué es Spring Framework y por qué usarlo. Recuperado 18 de marzo de 2022, de OpenWebinars.net website: <https://openwebinars.net/blog/conoce-que-es-spring-framework-y-por-que-usarlo/>
- Pacienza, E. (2018). *Metodologías de desarrollo de software*. 117.
- Peña Galbán, L. Y. (2018). Vínculo sociedad-universidad a través del programa de la disciplina Preparación para la Defensa. *Humanidades Médicas*, 14(3), 728-741.
- pmoinformatica.com, P. por. (2017). Requerimientos funcionales: Ejemplos. Recuperado 26 de octubre de 2022, de <http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>

Pressman, R. S. (2005a). *Ingeniería del Software: Un enfoque práctico*. McGraw-Hill Interamericana.

Pressman, R. S. (2005b). *Ingeniería del Software. Un Enfoque Practico 7ma edición*. Recuperado de https://www.academia.edu/24308956/Ingenieria_del_Software_Un_Enfoque_Practico_7ma_edici%C3%B3n

Pressman, R. S. (2010). *Ingeniería del software: Un enfoque práctico*. Recuperado de http://www.ingebook.com/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=4272

Quintero, L. (2020, junio 24). Qué son los Patrones de Diseño de software / Design Patterns. Recuperado 28 de octubre de 2022, de Profile Software Services website: <https://profile.es/blog/patrones-de-diseno-de-software/>

rédac, T. (2022, septiembre 2). IDE: ¿Qué es un Entorno de Desarrollo Integrado? Recuperado 24 de octubre de 2022, de Formation Data Science | DataScientest.com website: <https://datascientest.com/es/ide-que-es>

Reicek. (2017, agosto 16). PostgreSQL: Qué es, cómo funciona y cuáles son sus ventajas. Recuperado 24 de octubre de 2022, de Platzi website: <https://platzi.com/blog/que-es-postgresql/>

Solve, T. y. (2022, septiembre 26). Alta cohesión y bajo acoplamiento en diseño de software. Recuperado 28 de octubre de 2022, de Tech and Solve website:

<https://www.techandsolve.com/2022/09/alta-cohesion-y-bajo-acoplamiento-en-diseno-de-software/>

Sommerville, I. (2014). *Ingeniería de Software—Ian Sommerville 7ma Edición.pdf (PDFy mirror)*.

Recuperado de <http://archive.org/details/pdfy-g7E-3nrc3Q4yRJ4y>

UCI. (2014). Modifying an Approved AUP. Recuperado 12 de noviembre de 2022, de UCI Office of Research website: <https://research.uci.edu/animal-care-and-use/protocol-submission/rms-user-guide/modifying-an-approved-aup/>

UNIR. (2021, agosto 19). Framework: Qué es, para qué sirve y algunos ejemplos. Recuperado 2 de noviembre de 2022, de Edix España website:

<https://www.edix.com/es/instituto/framework/>

Urdaneta. (1990). Gestión de la Información—EcuRed. Recuperado 24 de octubre de 2022, de https://www.ecured.cu/Gesti%C3%B3n_de_la_Informaci%C3%B3n

Volodina. (2019). (PDF) UML y patrones | Ludmila Volodina—Academia.edu. Recuperado 13 de noviembre de 2022, de https://www.academia.edu/3105204/UML_y_patrones

Anexos

Tabla 10: HU: Autenticar usuario

Historia de usuario	
Número: 1	Requisito: RF Autenticar usuario
Usuario:	Programador: Frank A. Valero López
Prioridad en negocio: Alta	Riesgo en desarrollo: <ul style="list-style-type: none"> • Problemas de fallos eléctricos. • Planificación incorrecta.
Tiempo estimado:	Iteración asignada: 1
Descripción: El usuario introduce su usuario y contraseña, si los datos son válidos el usuario queda autenticado con el nivel de privilegios asignados de acuerdo a su rol; si los datos no son válidos el sistema muestra un mensaje de error.	
Observación:	
Prototipo elemental de interfaz gráfica de usuario:	

Tabla 11: HU: Modificar datos del usuario

Historia de usuario	
Número: 3	Requisito: RF3 Modificar datos del usuario
Usuario:	Programador: Frank A. Valero López
Prioridad en negocio: Alta	Riesgo en desarrollo:

	<ul style="list-style-type: none"> • Problemas de fallos eléctricos. • Planificación incorrecta.
Tiempo estimado: 2 días	Iteración asignada: 1
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir modificar datos de los usuarios en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para modificar los datos de un usuario hay que:</p> <ul style="list-style-type: none"> • Tener en cuenta los siguientes datos: nombre, apellidos, teléfono, dirección, grado militar, categoría, año académico, brigada, reservista, cargo militar, sexo, ci. • Estar autenticado en el sistema con el rol VDER. • Debe existir al menos en el sistema un usuario. <p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p> <p>Los campos nombre, apellidos, categoría y sexo son obligatorios.</p> <ul style="list-style-type: none"> • Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres. • Apellidos: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres. 	

- Categoría: Combobox con las opciones seleccionables de estudiante o profesor.
- Sexo: Combobox con las opciones seleccionables de masculino o femenino
- Reservista: Combobox con las opciones seleccionables de Si o No.

4- Flujo de la acción a realizar:

- El sistema debe permitir modificar un usuario, esta acción puede realizarse seleccionando la opción editar en el listado de usuarios.
- Cuando el usuario modifica de forma correcta los datos necesarios y selecciona la opción Actualizar, se muestra un mensaje de información de que el usuario fue modificado de forma correcta.
- Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al usuario de realizar nuevamente la acción en cuestión.
- Si selecciona la opción Cancelar regresará a la vista previa.

Observación:

Prototipo elemental de interfaz gráfica de usuario:

Tabla 12: HU: Eliminar usuario

Historia de usuario	
Número: 4	Requisito: RF4 Eliminar Usuario
Usuario:	Programador: Frank A. Valero López
Prioridad en negocio: Alta	Riesgo en desarrollo:

	<ul style="list-style-type: none"> • Problemas de fallos eléctricos. • Planificación incorrecta.
Tiempo estimado: 2 días	Iteración asignada: 1
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir eliminar un usuario en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para eliminar un usuario hay que:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema con el rol VDER. • Debe existir en el sistema al menos un usuario. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> • El sistema permite eliminar uno o varios usuarios, para eliminar varios usuarios, el sistema muestra un listado de los mismos permitiendo marcar más de un elemento a eliminar, selecciona la opción eliminar de las acciones en lote y presiona la opción Aplicar. • En caso de querer eliminar un elemento se puede realizar seleccionando la opción eliminar de las opciones que muestra el propio elemento o desde la vista previa del mismo. Para ambos casos el sistema muestra un mensaje de confirmación para Aceptar o no la acción que se está realizando. 	

<ul style="list-style-type: none"> • Si selecciona la opción Aceptar se actualiza el listado de usuarios y el sistema muestra un mensaje de información.
Observación:
Prototipo elemental de interfaz gráfica de usuario:

Tabla 13: HU: Listar usuarios

Historia de usuario	
Número: 5	Requisito: RF5 Listar Usuarios
Usuario:	Programador: Frank A. Valero López
Prioridad en negocio: Alta	Riesgo en desarrollo: <ul style="list-style-type: none"> • Problemas de fallos eléctricos. • Planificación incorrecta.
Tiempo estimado: 3 días	Iteración asignada: 1
Descripción: 1- Objetivo: <ul style="list-style-type: none"> • Permitir listar los datos de los usuarios en el sistema. 2- Acciones para lograr el objetivo (precondiciones y datos): Para incluir un nuevo usuario hay que: <ul style="list-style-type: none"> • Estar autenticado en el sistema con el rol VDER. 	

- Debe existir en el sistema al menos un usuario.

3- Flujo de la acción a realizar:

- Cuando se selecciona la opción Listar usuarios, aparecen todos los usuarios registrados en el sistema. Además, tiene la posibilidad de ver detalles de un usuario, modificarlo y eliminarlo.

Observación:

Prototipo elemental de interfaz gráfica de usuario:

Tabla 14: HU: Filtrar usuarios

Historia de usuario	
Número: 6	Requisito: RF6 Filtrar Usuarios
Usuario:	Programador: Frank A. Valero López
Prioridad en negocio: Alta	Riesgo en desarrollo: <ul style="list-style-type: none"> • Problemas de fallos eléctricos. • Planificación incorrecta.
Tiempo estimado: 2 días	Iteración asignada: 3
Descripción: 1- Objetivo: <ul style="list-style-type: none"> • Permitir filtrar un listado de usuarios en el sistema. 2- Acciones para lograr el objetivo (precondiciones y datos):	

<p>Para filtrar un listado de usuarios hay que:</p> <ul style="list-style-type: none"> • Tener en cuenta los siguientes datos: nombre, apellidos, categoría, sexo y reservista. • Estar autenticado en el sistema con el rol VDER. • Debe existir en el sistema al menos un usuario. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> • Cuando el usuario inserta un criterio de búsqueda y selecciona la opción buscar, el sistema muestra un listado de los elementos usuario que cumplen con los criterios establecidos, en caso de no encontrar ningún elemento el sistema muestra un mensaje de información.
Observación:
Prototipo elemental de interfaz gráfica de usuario:

Tabla 15: HU: Crear actividades para el aseguramiento del bastión

Historia de usuario	
Número: 7	Requisito: RF7 Crear actividades para el aseguramiento del bastión
Usuario:	Programador: Frank A. Valero López
Prioridad en negocio: Alta	Riesgo en desarrollo: <ul style="list-style-type: none"> • Problemas de fallos eléctricos. • Planificación incorrecta.
Tiempo estimado: 3 días	Iteración asignada: 2

Descripción:**1- Objetivo:**

Permitir crear nuevas actividades de aseguramiento para el bastión.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para crear nuevas actividades de aseguramiento del bastión hay que:

- Tener en cuenta los siguientes datos: actividad, fecha, hora, participantes, responsables, lugar.
- Estar autenticado en el sistema con el rol VDER.

3- Comportamientos válidos y no válidos (flujo central y alternos):

Los campos actividad, fecha, hora, participantes, responsables, lugar son obligatorios.

- Actividad: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 1000 caracteres.
- Fecha: campo de selección de fecha que contiene un calendario que permite elegir la fecha deseada.
- Hora: campo de selección de hora con una serie de elementos predefinidos.
- Participantes: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 250 caracteres.
- Responsables: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 500 caracteres y la posibilidad de añadir varios responsables.

- Lugar: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 250 caracteres.

4- Flujo de la acción a realizar:

- El sistema debe permitir incluir y/o seleccionar los datos para incluir una nueva actividad para el aseguramiento del bastión.
- Cuando se incluye y/o selecciona correctamente los datos necesarios para incluir una nueva actividad para el aseguramiento del bastión y se selecciona la opción Añadir, se crea un nuevo elemento y el sistema muestra un mensaje de información.
- Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al usuario de realizar nuevamente la acción en cuestión.
- Si selecciona la opción Cancelar regresará a la vista previa.

Observación:

Prototipo elemental de interfaz gráfica de usuario:

Tabla 16: HU: Modificar actividades para el aseguramiento del bastión

Historia de usuario	
Número: 8	Requisito: RF8 Modificar actividades para el aseguramiento del bastión.
Usuario:	Programador: Frank A. Valero López
Prioridad en negocio: Alta	Riesgo en desarrollo: <ul style="list-style-type: none"> • Problemas de fallos eléctricos. • Planificación incorrecta.

Tiempo estimado: 2 días	Iteración asignada: 2
Descripción:	
1- Objetivo:	
Permitir modificar datos de las actividades para el aseguramiento del bastión en el sistema.	
2- Acciones para lograr el objetivo (precondiciones y datos):	
Para modificar los datos de una actividad para el aseguramiento del bastión hay que:	
<ul style="list-style-type: none">• Tener en cuenta los siguientes datos: actividad, fecha, hora, participantes, responsables, lugar.• Estar autenticado en el sistema con el rol VDER.• Debe existir al menos en el sistema una actividad para el aseguramiento del bastión.	
3- Comportamientos válidos y no válidos (flujo central y alternos):	
Los campos actividad, fecha, hora, participantes, responsables, lugar son obligatorios.	
<ul style="list-style-type: none">• Actividad: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 1000 caracteres.• Fecha: campo de selección de fecha que contiene un calendario que permite elegir la fecha deseada.• Hora: campo de selección de hora con una serie de elementos predefinidos.• Participantes: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 250 caracteres.	

- Responsables: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 500 caracteres y la posibilidad de añadir varios responsables.
- Lugar: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 250 caracteres.

4- Flujo de la acción a realizar:

- El sistema debe permitir modificar una actividad para el aseguramiento del bastión, esta acción puede realizarse seleccionando la opción editar en el listado de actividades para el aseguramiento del bastión.
- Cuando el usuario modifica de forma correcta los datos necesarios y selecciona la opción Actualizar, se muestra un mensaje de información de que el usuario fue modificado de forma correcta.
- Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al usuario de realizar nuevamente la acción en cuestión.
- Si selecciona la opción Cancelar regresará a la vista previa.

Observación:

Prototipo elemental de interfaz gráfica de usuario:

Tabla 17: HU: Eliminar actividades para el aseguramiento del bastión

Historia de usuario	
Número: 9	Requisito: RF9 Eliminar actividades para el aseguramiento del bastión.
Usuario:	Programador: Frank A. Valero López
Prioridad en negocio: Media	Riesgo en desarrollo:

	<ul style="list-style-type: none"> • Problemas de fallos eléctricos. • Planificación incorrecta.
Tiempo estimado: 2 días	Iteración asignada: 2
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir eliminar una actividad para el aseguramiento del bastión del sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para eliminar una actividad para el aseguramiento del bastión hay que:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema con el rol VDER. • Debe existir en el sistema al menos una actividad para el aseguramiento del bastión. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> • El sistema permite eliminar una o varias actividades para el aseguramiento del bastión, para eliminar varias actividades para el aseguramiento del bastión, el sistema muestra un listado de las mismas permitiendo marcar más de un elemento a eliminar, selecciona la opción eliminar de las acciones en lote y presiona la opción Aplicar. • En caso de querer eliminar un elemento se puede realizar seleccionando la opción eliminar de las opciones que muestra el propio elemento o desde la vista previa del mismo. Para ambos casos el sistema muestra un mensaje de confirmación para Aceptar o no la acción que se está realizando. 	

<ul style="list-style-type: none"> • Si selecciona la opción Aceptar se actualiza el listado de usuarios y el sistema muestra un mensaje de información.
Observación:
Prototipo elemental de interfaz gráfica de usuario:

Tabla 18: HU: Consultar datos de unidad militar

Historia de usuario	
Número: 10	Requisito: RF19 Consultar datos de unidad militar
Usuario:	Programador: Frank A. Valero López
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Iteración asignada: 3
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir consultar los datos de una unidad militar en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para ver detalles de una unidad militar hay que:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema con el rol VDER. • Debe existir al menos en el sistema un usuario. 	

<p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p> <ul style="list-style-type: none"> • Debe haberse creado una unidad militar con anterioridad, de la misma pueden visualizarse los siguientes datos: nombre de unidad, listado de reservistas pertenecientes a dicha unidad. <p>4- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> • Inicialmente se muestra al usuario un listado con las unidades militares que han sido incluidas al sistema. Una vez seleccionada una unidad, podrá ver sus datos pulsando en la opción: “Visualizar”.
Observación:
Prototipo elemental de interfaz gráfica de usuario:

Tabla 19: HU: Generar reporte de informe resumen de la actividad del bastión

Historia de usuario	
Número: 11	Requisito: Generar reporte de informe resumen de la actividad del bastión.
Usuario:	Programador: Frank A. Valero López
Prioridad en negocio: Alta	Riesgo en desarrollo: <ul style="list-style-type: none"> • Problemas de fallos eléctricos. • Planificación incorrecta.
Tiempo estimado: 4 días	Iteración asignada: 3
Descripción:	

1- Objetivo:

Permitir generar reporte de informe resumen de la actividad del bastión en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para generar reporte de informe resumen de la actividad del bastión hay que:

- Estar autenticado en el sistema con el rol VDER.
- Debe existir al menos en el sistema un informe resumen de la actividad del bastión.

3- Flujo de la acción a realizar:

Cuando el usuario selecciona la opción generar reporte se muestra un listado con los siguientes datos:

- Asistencia a la actividad bastión universitario.
- Listado de actividades de preparación y realización.
- Listado de aspectos negativos y positivos observados durante la realización de la actividad bastión universitario.

Observación:**Prototipo elemental de interfaz gráfica de usuario:**

Descripción general: Permitir eliminar una actividad para el aseguramiento del bastión.

Condiciones de ejecución: Para eliminar una actividad para el aseguramiento del bastión hay que:

- Estar autenticado en el sistema con el rol VDER.

- Debe existir en el sistema al menos una actividad para el aseguramiento del bastión.

SC 1 Eliminar actividad para el aseguramiento del bastión.

Tabla 20 : Diseño de caso de caso de prueba para RF Eliminar actividad para el aseguramiento del bastión

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Eliminar actividad para el aseguramiento del bastión.	Selecciona la opción de eliminar una actividad para el aseguramiento del bastión.	Muestra un mensaje de confirmación. Y permite: - eliminar. - Cancelar.	Plan Bastión/Eliminar
EC 1.2 Opción Eliminar	Selecciona la opción Eliminar.	Elimina la actividad para el aseguramiento del bastión. Regresa al listado de las actividades para el aseguramiento del bastión actualizado y muestra un mensaje de información: "Actividad eliminada".	Plan Bastión/Eliminar/Eliminar
EC 1.3 Opción de cancelar.	Selecciona la opción de Cancelar.	Regresa a la vista anterior.	Plan Bastión/Eliminar/Cancelar

SC 2 Seleccionar acciones en lote

Tabla 21: Diseño de caso de prueba para acción Seleccionar acciones en lote

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 2.1 Opción de eliminar acciones en lote	Selecciona una o varias actividades de aseguramiento	Muestra un mensaje de confirmación. Y permite:	Plan Bastión/Eliminar

	para el bastión para aplicarle la acción de eliminar que aparece en lote.	- Eliminar. - Cancelar.	
EC 1.2 Opción Eliminar	Selecciona la opción Eliminar.	Elimina las actividades para el aseguramiento del bastión seleccionadas. Actualiza el listado de actividades para el aseguramiento del bastión y muestra un mensaje de información.	Plan Bastión/Eliminar/Eliminar
EC 1.3 Opción de cancelar.	Selecciona la opción de Cancelar.	Regresa al listado de instituciones.	Plan Bastión/Eliminar /Cancelar

Descripción general: Permitir modificar los datos de las actividades para el aseguramiento del bastión.

Condiciones de ejecución: Para incluir una nueva actividad para el aseguramiento del bastión hay que:

- Estar autenticado en el sistema con el rol VDER.
- Debe existir en el sistema al menos una actividad para el aseguramiento del bastión.

SC 1 Modificar Actividad para el aseguramiento del bastión.

Tabla 22: Diseño de caso de prueba para RF Modificar actividades para el aseguramiento del bastión

Escenario	Descripción	Actividad	Fecha	Hora	Participantes	Responsables	Lugar	Respuesta del sistema	Flujo central
EC 1.1 Opción	Selecciona la opción de modificar una							Muestra los datos de la institución seleccionada, per-	Plan Bastión/Modificar.

modificar datos	actividad para el aseguramiento para el bastión.							<p>mitiendo modificar los valores: - Actividad – Fecha – Hora - Participantes – Responsables - Lugar.</p> <p>Además: Permite: - Actualizar los datos. - Cancelar la operación en cualquier momento.</p>	
EC 1.2 Opción de actualizar los datos.	Modifica los datos que necesite y selecciona la opción de actualizar los datos de la actividad de aseguramiento.	V	V	V	V	V	V	<p>Valida los datos. Actualiza la actividad para el aseguramiento del bastión. Muestra el listado de actividades para el aseguramiento del bastión y un mensaje de información: "Actividad modificada".</p>	Plan Bastión/Modificar/Añadir
EC 1.3 Opción de cancelar.	Selecciona la opción de Cancelar.							<p>Elimina los datos creados. Regresa al listado de actividades para el aseguramiento del bastión.</p>	Plan Bastión/Modificar/Cancelar
EC 1.4 Datos incompletos	Existen datos incompletos.	I	V	V	V	V	V	<p>Deshabilita la opción añadir. Muestra un indicador sobre los campos vacíos y un mensaje: "El campo no puede estar vacío". <u>Regresa al EC 1.1.</u></p>	Plan Bastión/Modificar/Añadir
		V	I	V	V	I	V		
		V	V	I	V	V	V		
EC 1.5 Datos incorrectos	Existen datos incorrectos.	V	V	V	I	V	V	<p>Muestra un mensaje de información. Muestra un indicador sobre los campos incorrectos. <u>Regresa al EC 1.1.</u></p>	Plan Bastión/Modificar/Añadir
		V	V	V	V	I	V		