

UNIVERSIDAD DE GUANTÁNAMO
Facultad de Ingeniería y Ciencias Técnicas

*Sistema informático para la gestión del trabajo operativo y
pericial en la Criminalística (SISOPC).*

Tesis presentada en opción al Título de Ingeniero Informático.

Autor: Biolkis Quintero Delgado

Tutores: Ing. Yelene Marcelo Luis

Ing. Yilian Rodríguez Grille

Guantánamo, julio 2020

Cuba

DEDICATORIA.

A:

Dios por darnos la oportunidad de vivir y regalarnos una familia maravillosa.

Mis padres por ser fuente de inspiración y deseo de superación, por apoyarnos en los malos momentos y alentarnos en los buenos.

Mi madre por brindarme su amor y comprensión. (Gracias por todo el apoyo).

Familiares por el apoyo y por estar siempre a mi lado compartiendo todos los momentos, los quiero...

De forma especial a Yelene Marcelo Luis, Eudel Campuzano, Haydee Herrera y Roberto Lambert por dedicarme tanto tiempo y asesoría constante.

AGRADECIMIENTOS

Asesores de tesis y colegas del órgano, Eudel, Haydee y Yelene:

Por su amistad sincera y apoyo brindado en la realización del trabajo de tesis, y así mismo, por compartir sus conocimientos desinteresadamente, por su esfuerzo y colaboración sin escatimar días ni horas que le restaban a su descanso y vida familiar.

A la Universidad de Guantánamo y en especial a la Facultad de Ingeniería y Ciencias Técnicas por abrirnos las puertas y brindarnos la oportunidad de convertirnos en profesionales.

Nuestros amigos y compañeros:

Por compartir todos los momentos gratos y acompañarnos a lo largo de nuestro recorrido.

A todos los trabajadores del Órgano de informática Comunicaciones y Cifras por toda la ayuda que nos brindaron durante la realización de este trabajo de diploma.

A mis familiares y en especial a Biolkis Quintero y Dania Delgado por ser los guías de mi vida y por todo su esfuerzo, por lograr que me hiciera un profesional: gracias por ser los mejores padres del mundo.

RESUMEN

La presente investigación tiene como objetivo contribuir a elevar la gestión del trabajo operativo y pericial, llevado a cabo en el Órgano de Criminalística del Ministerio del Interior mediante el desarrollo de un sistema informático. Para el desarrollo de la investigación se utilizaron los métodos teóricos: análisis y síntesis, inducción y deducción, modelación; como métodos empíricos: entrevistas y análisis de documentos. Como metodología para la elaboración de la propuesta se utilizó Scrum. La aplicación fue desarrollada utilizando el lenguaje de programación PHP, las tecnologías HTML, CSS y Java Script, el marco de trabajo Bootstrap y el gestor de base de datos Oracle. Para la validación se utilizaron pruebas de caja negra. Como resultado se obtuvo una aplicación web que facilita la gestión de la información concerniente a los procesos realizados en las salidas operativas.

Palabras Claves: Criminalística; Salidas Operativas; Trabajo Operativo y Pericial; Ministerio del Interior; Lugar del Hecho; Órganos.

ABSTRACT

The objective of this investigation is to contribute to the management of operational and expert work, carried out in the Criminalistics Organ of the Ministry of the Interior through the development of a computer system. For the development of the investigation the theoretical methods were used: analysis and synthesis, induction and deduction, modeling; as empirical methods: interviews and document analysis. Scrum was used as the methodology for preparing the proposal. The application was developed using the PHP programming language, HTML, CSS and Java Script technologies, the Bootstrap framework and the Oracle database manager. Black box tests were used for validation. As a result, a web application was obtained that facilitates the management of information concerning the processes carried out in the operational exits.

Keywords: Criminalistics; Operational outputs; Operative and Expert Work; Ministry of Interior; Place of Fact; Organs.

Índice

INTRODUCCIÓN.....	1
CAPÍTULO 1: DESCRIPCIÓN DEL PROCESO GESTIÓN DEL TRABAJO OPERATIVO Y PERICIAL EN EL ÓRGANO DE CRIMINALÍSTICA DEL MININT EN GUANTÁNAMO.....	5
1.1. Caracterización del proceso gestión del trabajo operativo y pericial.....	5
1.2 Descripción de las tendencias y tecnologías actuales.....	8
1.2.1 Metodología utilizada: SCRUM.....	8
1.2.2. Sistemas de Gestión de Bases de Datos (SGBD)	10
1.2.3 Lenguajes de programación.....	11
1.2.4 Marcos de trabajos.....	14
1.2.5. Servidor web.....	14
1.2.6 Arquitectura.....	15
1.2.7 Patrón arquitectónico.....	16
1.2.8 Herramienta CASE de planificación.....	17
1.3. Descripción del proceso de negocio que será objeto de informatización.....	18
1.3.1 Reglas del negocio.....	18
Conclusiones parciales.....	19
CAPITULO 2: DESARROLLO DEL SISTEMA INFORMÁTICO PARA LA GESTIÓN DEL TRABAJO OPERATIVO Y PERICIAL EN LA CRIMINALISTICA.....	20
2.1 Especificación de los requisitos de software.....	20
2.1.1 Requerimientos funcionales.....	20
2.1.2 Requerimientos no funcionales.....	21
2.2 Descripción del Sistema.....	23
2.2.1 Historias de usuarios para los casos críticos.....	23
2.2.2 Sprint Backlog.....	24
2.3 Análisis del sistema.....	26
2.3.1 Diagrama de comunicación.....	26
2.4 Diseño del Sistema.....	28
2.4.1 Patrones de diseño.....	29

2.4.2 Descripción de la arquitectura del software.....	30
2.4.3 Modelo de Entidad Relación.	31
2.5 Implementación.	32
2.5.1 Diagrama de despliegue.	32
2.6 Diseño de caso de prueba.	34
2.6.1 Prototipos de la interfaz de usuario para los casos críticos.....	35
2.6.2 Prueba de caja negra.	35
2.7 Estudio de factibilidad.....	37
2.7.1 Planificación basada en casos de uso.	38
Conclusiones parciales.....	48
CONCLUSIONES GENERALES.....	49
RECOMENDACIONES.....	50
BIBLIOGRAFÍA.....	51

Índice de tablas

Tabla 1: Crear Planilla del trabajo operativo del perito en el lugar del suceso.	23
Tabla 2: Enviar planilla a la (OCDP).....	24
Tabla 3: Exportar planilla.	24
Tabla 4: Gestionar Roles de Usuario, Gestionar Usuarios, Autenticar usuario	25
Tabla 5: Sprint Backlog: Mostrar información de las salidas operativas.	26
Tabla 6: Servidor de Base de Datos.....	33
Tabla 7: Cliente PC.....	34
Tabla 8: Servidor Web.	34
Tabla 9: Diseño de prueba de caja negra del caso de prueba: Comprobar que no se acceda al sistema con un usuario y una contraseña incorrecta.	36
Tabla 10: Mostrar Salidas Operativas.	37
Tabla 11: Factor de peso de los actores.	38
Tabla 12: Factor de peso de los casos de uso	39
Tabla 13: Factor de complejidad técnica.	41
Tabla 14: Factor Ambiente.....	42
Tabla 15: Esfuerzo por cada actividad del proyecto.	43
Tabla 16: Cuantificación de los beneficios tangibles.	45
Tabla 17: Relación de costos en los que se incurren antes y después del sistema.	46
Tabla 18: Relación de costos teniendo en cuenta la cantidad de órganos.	46
Tabla 19: Retorno de la Inversión	47

Índice de figuras

Figura 1. Ciclo de SCRUM– Sprint.....	10
Figura 2. Actividades del proceso de SCRUM	10
Figura 3. Diagrama de comunicación para:Crear planilla del trabajo operativo del perito en el lugar del suceso.....	27
Figura 4: Diagrama de comunicación para la Oficina de Control de Documentos Periciales.....	28
Figura 5. Patrón de diseño MVC.....	30
Figura 6: Modelo de Entidad Relación.	32
Figura 7: Diagrama de despliegue.	33
Figura 8: Punto de equilibrio.	46
Figura 9: Retorno de la inversión.	47

INTRODUCCIÓN

Criminalística, es la disciplina que aplica fundamentalmente los conocimientos, métodos y técnicas de investigación de las ciencias naturales en el examen del material sensible significativo, relacionado con un presunto hecho delictuoso con el fin de determinar en auxilio de los órganos encargados de administrar justicia, su existencia o bien reconstruirlo o bien señalar y precisar la intervención de uno o varios sujetos en el mismo. (Moreno, op.cit, págs. 344-345).

En cuanto a sus principios, la criminalística nos capacita para conocer la verdad de las cosas, materia de sus estudios y luego nos adiestra para hacer bien tales cosas. Para llegar a esa verdad, se impone la ejecución de los siguientes procedimientos:

- Comprobación
- Sistematización
- Objetividad

Se puede decir que la Criminalística no se limita al recojo de indicios o la evidencia que cae en su dominio, sino que estos han de ser sometidos a diversos experimentos y comprobaciones. El desarrollo de esta labor, así como de sus resultados por más justificados que están, obedecerán a una organización metódica y sistematizada; finalmente el saber científico de la Criminalística, en cuanto a su objetividad se pone al servicio de la administración de justicia, para combatir el delito, contribuyendo a su esclarecimiento y señalando a los autores. (Criminalística de Laboratorio, Dr. Romualdo Ayala Ponce, pág. 7)

Actualmente a nivel internacional existen muchas especialidades de criminalística, para realizar los correspondientes procesos de gestión utilizan sistemas informáticos. En Cuba, en el Órgano de Criminalística no hay presente ningún sistema que administre o gestione las salidas operativas de dicho órgano.

En este contexto el Órgano de Criminalística del Ministerio del Interior (MININT) en Guantánamo, realiza su gestión de las salidas operativas de una forma constante, pero mediante soporte de papel, por lo que el Órgano de Criminalística del MININT en Guantánamo propone informatizar las salidas operativas realizadas por los peritos e instructores, para que sea más efectivo el proceso de consultar los resultados de las salidas operativas.

Mediante entrevistas realizadas a directivos de la entidad, se pudo constatar que durante varios años en el Órgano de Criminalística del MININT en Guantánamo, el trabajo pericial podría realizarse de forma más eficiente, pero la existencia de varias deficiencias no lo hacía posible, tales como: la falta de organización de la información, la demora en la asignación del trabajo, la

tardanza en la entrega de los informes, la actualización de los datos requiere de un elevado tiempo debido a la dispersión de la información y duplicidad recopilada, así como el poco control de la información debido al trabajo mediante soporte en papel, resultaron ser las más notables dificultades que se presentaron. Por lo que la búsqueda de una vía de solución para lograr una mejor administración de las salidas operativas pasó a ocupar un lugar importante en el banco de problemas de la institución.

En el Órgano de Criminalística del MININT en Guantánamo se utilizan varios sistemas y de todos necesitan datos para poder obtener una estadística al final de un trabajo pericial.

Identificadas las deficiencias se definió como **problema a resolver**: ¿Cómo elevar la eficiencia en el proceso gestión del trabajo operativo y pericial en el Órgano de Criminalística del MININT en Guantánamo?

Para lograr una mejor comprensión del problema, se tiene como **objeto de estudio**: el proceso gestión del trabajo operativo y pericial.

Para dar solución al problema anteriormente planteado se definió como **objetivo**: desarrollar un sistema informático que permita elevar la eficiencia en el proceso gestión del trabajo operativo y pericial en el Órgano de Criminalística del MININT en Guantánamo.

El **campo de acción** de la investigación se centrará en el manejo de la información de las salidas operativas en el Órgano de Criminalística del MININT en Guantánamo.

Se plantea como **idea a defender**: el desarrollo de un sistema informático contribuirá a elevar la eficiencia en la gestión del trabajo operativo y pericial en el Órgano de Criminalística del MININT en Guantánamo.

Para cumplir los objetivos propuestos se elaboraron las siguientes **tareas de investigación**:

1. Caracterizar el proceso gestión del trabajo operativo y pericial en el Órgano de Criminalística del MININT en Guantánamo.
2. Estudiar y seleccionar las tecnologías y herramientas actuales utilizadas en el desarrollo del sistema informático propuesto.
3. Realizar el análisis y diseño de la solución propuesta.
4. Implementar el sistema propuesto.
5. Realizar las pruebas de caja negra con el objetivo de detectar errores antes de poner la aplicación en ejecución.

6. Realizar un estudio de factibilidad, que permita evaluar los esfuerzos en la realización del sistema informático propuesto.

Para dar cumplimiento a las tareas antes mencionadas se aplicaron los siguientes **métodos de investigación**.

Métodos Teóricos:

- Análisis y síntesis: se utilizó para lograr una adecuada comprensión sobre las tendencias del objeto que se investiga, así como la identificación del problema, elaboración de los fundamentos teóricos y la formulación de la propuesta de solución, también para procesar e interpretar la documentación referente a las etapas de análisis y diseño del sistema.
- Histórico y Lógico: se emplea para profundizar la trayectoria de la gestión del trabajo operativo y pericial en el Órgano de Criminalística del MININT en Guantánamo.
- Modelación: fue utilizado para diseñar una aplicación adecuada a las características de los técnicos y las condiciones en el proceso gestión del trabajo operativo y pericial en el Órgano de Criminalística del MININT en Guantánamo.

Métodos Empíricos:

- Observación Científica: se observa el desarrollo del proceso gestión del trabajo operativo y pericial que se realiza en la criminalística para detectar las deficiencias y diagnosticar su estado.
- Entrevista: se aplica para obtener información, opiniones y criterios de instructores y peritos que intervienen en el proceso gestión del trabajo pericial.
- Análisis de documentos: se realizó la revisión de documento digitales y los libros de registros de entrada y salida que recoge los datos de las salidas operativas realizadas para comprender su funcionamiento y conformar los posibles requerimientos funcionales.

La investigación presentada está dividida en dos capítulos:

Capítulo 1: presenta la caracterización del proceso gestión del trabajo operativo y pericial que se lleva a cabo en las salidas operativas. Se describen las reglas que se deben cumplir para el desarrollo de este. Además, se justifican las herramientas y la metodología utilizada para la propuesta presentada.

Capítulo 2: este capítulo presenta la solución propuesta a partir del desarrollo de la metodología presentada para la implementación del sistema en el perfeccionamiento del

trabajo pericial. También se especifican los requerimientos funcionales y no funcionales del sistema, se definen las historias de usuario; se hace referencia al análisis y diseño del sistema, se realiza el diseño de casos de prueba y el estudio de factibilidad a la aplicación.

CAPÍTULO 1: DESCRIPCIÓN DEL PROCESO GESTIÓN DEL TRABAJO OPERATIVO Y PERICIAL EN EL ÓRGANO DE CRIMINALÍSTICA DEL MININT EN GUANTÁNAMO.

Este capítulo abordará las descripciones sobre el proceso gestión del trabajo operativo y pericial, se evalúa el estado actual del proceso estudiado. Se realizará una breve descripción de las tecnologías de desarrollo utilizadas, los principales lenguajes de programación, sistema gestor de base de datos y algunas herramientas utilizadas para darle solución al problema.

1.1. Caracterización del proceso gestión del trabajo operativo y pericial.

La criminalística, como rama de la ciencia se encarga con el apoyo de otras ciencias de la investigación de las huellas, muestras e indicios que son captados en el lugar del hecho mediante inspección dinámica del perito, que se redactada en acta bajo la tutela legal de la Ley 5: Ley de Procedimiento Penal, la misma se divide como forma organizativa en tres unidades, el Laboratorio Provincial de Criminalística que es el lugar donde son procesadas todas la pruebas materiales extraídas y trasladadas desde el lugar del hecho, la Unidad de Enfrentamiento, que es la encargada de asistir al lugar del hecho para el levantamiento de todo el material probatorio de vinculación con los sospechosos de los casos trabajados y la Unidad Básica de la Criminalística, unidad que radica en el municipio de Baracoa, acoge a los demás municipios del sur de la provincia y se encarga de enfrentar todos los hechos de su competencia y el procesamiento de menor complejidad de las huellas, muestras e indicios que son ocupadas en el lugar del hecho.

El perito como integrante de las Guardias Operativas, asiste a todos los hechos en que es convocado con el objetivo de buscar, revelar, fijar, extraer, embalar y trasladar cumpliendo lo establecido en la Orden 36 del Ministerio del Interior, todas las huellas, muestras e indicios del tipo dérmico (dactilar, palmar, podoral y kiloscopica). Además las muestras digitopalmares), química (sangre, orina, suelo, pintura, lámpara de vehículos, drogas, muestras viscerales, tabletas, grajeas y comprimidos, armas de fuego, papel, tintas y colorantes, derivados del petróleo, bloques, chasis y carrocerías de vehículos, sustancia orgánicas, inorgánicas, polímeros, metales y sus aleaciones, etc.), trazológicas (instrumentos de corte simple, por encuentro y cizallamiento, objetos con huellas de corte, objetos fracturados, calzados y huellas de calzado, sellos, huellas de arcada dental, huellas de transporte, etc.), documentológicas (documentos manuscritos, impresos, mecanografiados, seguridad, papel moneda y similares, impresoras, tintas, papel, bolígrafos, etc.), informáticas (todos los dispositivos de infocomunicaciones que almacene datos, teléfonos, tabletas, memorias, tarjetas, discos de computadoras, cámaras fotográficas, etc.), del AVEXI (elementos relacionados con las averías, las explosiones y los incendios), biológicas (sangre, semen, saliva, otros fluidos y secreciones

corporales, pelos, fibras y micro fibras, restos óseos, material vegetal, tejido visceral, muscular y dérmico, piezas dentales, etc.), balísticas (casquillos, proyectiles y armas de fuego), odorológicas (huellas e impresiones olorosas ocupadas a sospechosos), del IPRE (fotografías, personas, restos óseos), fonoscópicas (grabaciones y medios de grabación); que a partir del momento en que son recogidas en acta, se convierten en pruebas materiales que debidamente descritas y embaladas se hacen llegar con su solicitud de peritaje a la Oficina de Control de Documentos Periciales (OCDP) para que estas a su vez las remita hacia cada disciplina que debe investigar de acuerdo a lo solicitado en cada prueba material que le compete y emitir un informe de los resultados alcanzados en un término no mayor a los quince días, desde su llegada a la OCDP.

El Laboratorio Provincial de Criminalística es el encargado de realizar de acuerdo con cada una de las metodologías aprobadas, las investigaciones desde el punto de vista Diagnóstico o Comparativas – Identificativas, de consonancia con las características de las huellas, indicios y muestras remitidas.

En un término de quince días cada disciplina del Laboratorio, estudiará y elaborará los dictámenes periciales ordinarios que podrán ser retirados en dependencia de la satisfacción del solicitante o complementarios por la conectividad de las investigaciones.

Una vez concluida la investigación de cada disciplina, devuelve el dictamen pericial con los anexos correspondientes a la Oficina de Control de Documentos Periciales (OCDP) y es en este sitio donde los órganos solicitantes recogen el resultado de sus solicitudes de investigación.

El asesoramiento legal y criminalístico de los Expedientes Investigativos del Control Interno (CI) y la Policía Técnica Investigativa (PTI), constituye la actividad de mayor importancia en materia de lograr el corte público de los procesos, los peritos realizan propuestas en función de lograr convertir la información operativa en elementos de prueba.

A su vez, la Unidad Básica de Criminalística (UBC) se encarga de concurrir a los casos de su competencia y realizar los peritajes de menor complejidad, fundamentalmente Peritajes Diagnósticos de las Disciplinas de Biología Criminalística y Trazología.

La Especialidad Criminalística en el MININT tiene como **Misión:** descubrir, obtener e investigar los indicios, huellas y muestras relacionadas con los delitos, así como determinar las vías, las circunstancias, modos y medios empleados para su comisión, demostrar, documentar y

legalizar los elementos de prueba en los procesos investigativos y de instrucción, la identificación de sus autores y enmascaramiento de las fuerzas, métodos y medios del Trabajo Operativo Secreto; y con fines preventivos determina los riesgos, causas y condiciones que favorecen su ocurrencia.

Su **Visión** es fortalecer las concepciones de trabajo, adquirir mayor eficacia, eficiencia y rigor técnico-profesional desarrollando un pensamiento investigativo, analítico e integral en los peritos a partir de la aplicación de la técnica, la metodología y la táctica utilizando los adelantos científico técnicos que contribuyan a la ejecución de acciones investigativas para la obtención de los indicios necesarios que permitan documentar y legalizar los elementos de pruebas suficientes del delito e identificación de autores.

Como objetivos estratégicos presenta:

Objetivo formal

Auxiliar con los resultados base de análisis técnico-científico, metodología y tecnología, a los órganos que cumplen funciones de policía judicial y a los que les corresponden administrar justicia, a efecto de darles elementos probatorios, identificadores y reconstructores conducentes a establecer la verdad de los hechos que investigan.

Objetivo material

El objetivo material de la criminalística hace relación al estudio de los elementos materiales de prueba que se utilizan y son producto de la comisión de los hechos.

Objetivo general

El estudio de las evidencias materiales en la investigación criminalística, tanto en el campo de los hechos, como en el laboratorio, llevan un objetivo general perfectamente definido y circunscrito a cinco tareas básicas e importantes.

- 1) Investigar técnicamente y demostrar científicamente, la existencia de un hecho en particular probablemente delictuoso.
- 2) Determinar los fenómenos y reconstruir el mecanismo del hecho, señalando los instrumentos u objetos de ejecución, sus manifestaciones y las maniobras que se pusieron en juego para realizarlo.
- 3) Aportar evidencias o coordinar técnicas o sistemas para la identificación de la víctima, si existiese.
- 4) Aportar evidencias para la identificación del o los presuntos imputados autores.

- 5) Y aportar las pruebas indiciarias para probar el grado de participación del o los presuntos autores y demás involucrados.

Deficiencias encontradas en el proceso de trabajo operativo y pericial:

- La asignación del trabajo pericial en ocasiones se demora, lo que genera tardanza ante solicitudes hechas por directivos.
- La actualización de los datos requiere de un elevado tiempo debido a la dispersión de la información y duplicidad de la misma.
- El control de la información sobre el trabajo pericial, se realiza de forma manual, lo que trae como consecuencias que haya mayor probabilidad de errores por el gran volumen de datos y que se requiera de mayor tiempo en el momento de obtener información.

1.2 Descripción de las tendencias y tecnologías actuales.

Un sistema informático, como todo sistema, es el conjunto de partes interrelacionadas, hardware, software y de recurso humano que permite almacenar y procesar información; refiriendo el término hardware a toda parte tangible, el término software al equipamiento lógico o soporte lógico y el recurso humano al equipo técnico que crea y mantiene dicho sistema, así como los usuarios que lo utilizan, [IEEE Std, 1993].

El creciente desarrollo de las tecnologías aumenta las probabilidades de elegir herramientas que satisfagan las necesidades de los clientes, evaluando su utilidad y eficacia práctica. Todo este constante movimiento es indispensable para el desarrollo del software. A continuación, se hace un estudio de las tendencias actuales de las tecnologías y selección de las herramientas más acertadas para cumplir los requerimientos de los clientes, por ejemplo: metodologías de desarrollo de software (MDS), lenguajes de programación, lenguaje de modelado y gestores de bases de datos, teniendo en cuenta las que mejores se adaptan a lo que se quiere realizar.

1.2.1 Metodología utilizada: SCRUM.

Es necesario contar con una metodología de desarrollo que se ajuste a las características del proyecto para garantizar la calidad y efectividad de software, sobre todo cuando el proyecto es de gran envergadura. En este caso, la metodología de desarrollo a utilizar es SCRUM. Es una metodología ágil que surge como alternativa a las tradicionales. Las metodologías ágiles proponen un marco de gestión de proyectos más pragmáticos y menos burocráticos.

Beneficios de SCRUM:

- Alineamiento entre cliente y equipo.
- Gestión regular de las expectativas del cliente.
- Resultados a corto plazo.
- Retorno de inversión (ROI).
- Equipo motivado.
- Flexibilidad y adaptación a los cambios.
- Calidad del producto final.

¿Por qué utilizar SCRUM?

Es un modo de desarrollo de carácter adaptable más que predictivo. Orientado a las personas más que a los procesos. Emplea la estructura de desarrollo ágil: incremental, basada en iteraciones y revisiones.

En resumen, lo que propone SCRUM es un marco de trabajo, un camino simple y claro con una serie de artefactos, roles y rituales que permiten transitarlo en base a iteraciones basadas en inspeccionar y adaptar donde cada vez que culmina se debe haber alcanzado un producto que en sí mismo tenga retorno de la inversión para el cliente.

En SCRUM un proyecto se ejecuta en bloques temporales (iteraciones-Sprint) de un mes natural (pueden ser de dos o tres semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto que sea susceptible de ser entregado con el mínimo esfuerzo cuando el cliente lo solicite.



Figura 1. Ciclo de SCRUM– Sprint

Las actividades que se plantea realizar en la metodología Scrum son las siguientes:

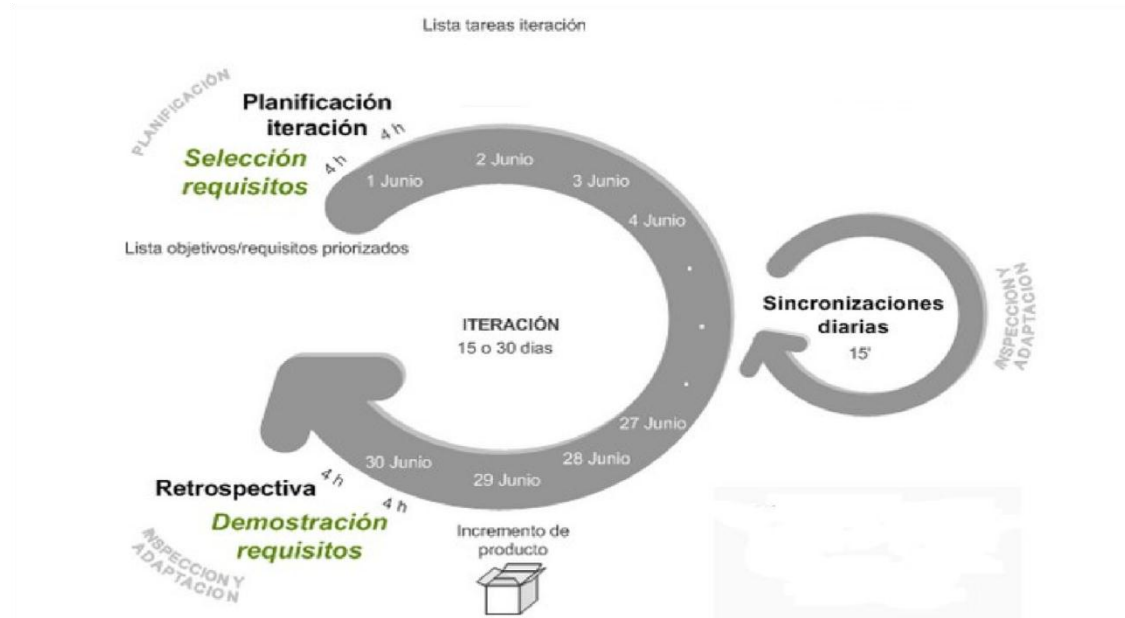


Figura 2. Actividades del proceso de SCRUM

1.2.2. Sistemas de Gestión de Bases de Datos (SGBD)

Los Sistemas de Gestión de Bases de Datos (Database Management System, SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Oracle 11g: Es un gestor de bases de datos fabricado por Oracle Corporation y es multiplataforma, disponible en Windows, Linux y Unix. El MININT Guantánamo lo utiliza para el desarrollo de sus aplicaciones convirtiéndose de esta manera en política de la institución.

¿Por qué usar Oracle?

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relacional Data Base Management System), desarrollado por Oracle Corporation.

Principales características:

- Entorno cliente / servidor.
- Gestión de grandes bases de datos.
- Usuarios concurrentes.

- Alto rendimiento en transacciones.
- Sistemas de alta disponibilidad.
- Disponibilidad controlada de los datos de las aplicaciones.
- Adaptación a estándares de la industria, como SQL-92.
- Gestión de la seguridad.
- Autogestión de la integridad de los datos.
- Opción distribuida.
- Portabilidad.
- Compatibilidad.
- Conectabilidad.
- Replicación de entornos.

Ventajas del Oracle 11g

- Oracle es el motor de base de datos relacional más usado a nivel mundial.
- Puede ejecutarse en todas las plataformas, desde una PC hasta un supercomputador.
- Oracle soporta todas las funciones que se esperan de un servidor "profesional": un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa bastante potente.
- Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.
- El software del servidor puede ejecutarse en multitud de sistemas operativos.
- Oracle es la base de datos con más orientación hacia INTERNET.
- Un aceptable soporte.

1.2.3 Lenguajes de programación.

Lenguajes del lado del cliente.

Son lenguajes que se ejecutan en el lado del cliente, los mismos son interpretados por el navegador generando finalmente páginas estáticas o dinámicas como respuesta, posibilitando en la mayoría de los casos mejorar la interfaz de usuario. A continuación, son expuestos los lenguajes del lado del cliente utilizados para la elaboración del sistema informático propuesto.

HTML: Es un lenguaje de marcado de contenido para la elaboración de páginas web. Es un estándar que en sus diferentes versiones define una estructura básica y un código para establecer el contenido de la página web, como texto, imágenes, formularios, controles, etc. El HTML basa su filosofía de desarrollo en la referenciación, técnica mediante la cual se añade cualquier elemento externo a las páginas como imagen y video, estos no se incrustan directamente en el código, sino que se hace una referencia a su ubicación, de modo que la tarea de unificar y visualizar la página final recae sobre el navegador. HTML puede incluir scripts (Java Script y PHP), además puede indicar cómo hacer un documento interactivo a través de ligas especiales de hipertexto, las cuales conectan diferentes documentos, así como otros recursos de Internet. Java Script.

CSS: Es un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas, como HTML. CSS tiene una sintaxis muy sencilla, compuesta por una lista de reglas, cada regla consiste en uno o más selectores y un bloque de declaración con los estilos a aplicar sobre los elementos de los documentos que cumplan con los selectores. El uso de CSS presenta un conjunto de ventajas como el control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo, además permite la separación del contenido de la presentación, lo que facilita al creador o diseñador la modificación de la visualización del documento sin alterar el contenido del mismo.

Java Script: Es un lenguaje de programación interpretado, es decir, que no requiere compilación, es imperativo, dinámico, orientado a objetos, basado en prototipos y multiplataforma, su código se incluye directamente en el mismo documento permitiendo mejoras en la interfaz de usuario. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es uno de los lenguajes de programación del lado del cliente más utilizado. Con JavaScript, gran parte de la programación está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros, permitiendo crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las

instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso con que cuenta este lenguaje es el propio navegador.

Lenguajes del lado del servidor.

Son lenguajes que se ejecutan en el lado del servidor, los mismos procesan las peticiones del usuario mediante la interpretación de un script en el servidor web, facilitando el acceso a las bases de datos y generando finalmente páginas HTML dinámicas como respuesta.

PHP

PHP es un Acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Es un lenguaje de programación, lenguaje interpretado, diseñado originalmente para la creación de Páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Entre sus principales ventajas se encuentran:

- Es un lenguaje multiplataforma. Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial (Sitio Oficial), entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida. No requiere definición de tipos de variables, aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

- Tiene manejo de excepciones (desde PHP5).
- Su inconveniente es que la ofuscación de código es la única forma de ocultar los códigos fuentes.

1.2.4 Marcos de trabajos.

En el desarrollo de software, un marco de trabajo es una estructura conceptual y tecnológica de soporte definido, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Por lo general, incluye soportes de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. (Gutiérrez, s.f)

Los framework hacen mucho más dinámico el desarrollo de aplicaciones web, poniendo a nuestra disposición estructuras básicas mediante las cuales construir las aplicaciones. En otras palabras, los framework ayudan a promover el rápido desarrollo de aplicaciones, y reducen la cantidad de código repetitivo para los desarrolladores.

Bootstrap: Es un framework web o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo

Los componentes de Java Script para Bootstrap están basados en la librería jQuery de Java Script. Los plug-ins se encuentran en la herramienta de plug-in de jQuery. Proveen elementos adicionales de interfaz de usuario como diálogos, tooltips y carruseles. También extienden la funcionalidad de algunos elementos de interfaz existentes, incluyendo por ejemplo una función de auto-completar para campos de entrada (input).

1.2.5. Servidor web.

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente es interpretado por un navegador web.

Apache: En la implementación del sistema informático propuesto se utilizó el servidor web apache, el mismo es posible de utilizar en distintas plataformas y entornos. Es altamente configurable de diseño modular, posibilitando que los administradores de sitios web puedan

elegir los módulos que serán incluidos y ejecutados en el servidor. (Kabir, 2004). Es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Apache es un servidor Web de código abierto, su diseño modular provee de un alto grado de calidad y fortaleza a las implementaciones que utilizan el protocolo HTTP. Trabaja con gran cantidad de lenguajes como Perl, PHP y otros lenguajes de script; además de proporcionar a los contenidos de información un lugar donde estar a disposición de forma segura y confiable.

Entre sus principales características destacan que es:

- Multiplataforma.
- Es un servidor web conforme al protocolo HTTP.
- Modular.
- Código abierto.
- Extensible.
- Popular (fácil conseguir ayuda/suporte)

1.2.6 Arquitectura.

La Arquitectura es la estructura de los componentes más significativos de un sistema interactuando a través de interfaces con otros componentes conformados por componentes sucesivamente pequeños e interfaces. (Jacobson, Rumbaugh, & Booch, 2006)

Arquitectura Cliente-Servidor: Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajos inteligentes o “clientes”, resultan en un trabajo realizado por otros computadores llamados “servidores”. La arquitectura cliente-servidor puede ser clasificada en dos capas, tres capas o n capas.

A continuación, se muestran algunas ventajas de la arquitectura cliente-servidor:

- Centralización del control: Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Escalabilidad: Se puede aumentar la capacidad de clientes y servidores por separado.

- Los cambios realizados en las plataformas de los clientes o de los servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- Las funciones de cliente y servidor pueden estar en plataformas separadas, o en la misma plataforma.
- Un servidor da servicio a múltiples clientes en forma concurrente.
- Cumplimiento de expectativas: El cliente establece sus expectativas indicando el valor que aporta a cada requisito / historia del proyecto, el equipo los estima y con esta información el propietario del producto establece su prioridad.
- Flexibilidad a cambios: Genera una alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado. La metodología está diseñada para adaptarse a los cambios de requerimientos que conllevan los proyectos complejos.
- Reducción del tiempo: El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo.
- Mayor calidad del software: La forma de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.

1.2.7 Patrón arquitectónico.

Los patrones arquitectónicos son estándares de diseño de software que ofrecen soluciones a problemas de arquitectura de software; dan una descripción de los elementos, y el tipo de relación que tienen, junto a un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural, esencial para un sistema de software que consta de subsistemas, responsabilidades e interrelaciones. (Jacobson et al., 2006)

Patrón arquitectónico a utilizar: MVC

Para la implementación del sistema informático propuesto se utilizó el patrón arquitectónico Modelo-Vista-Controlador (MVC), el cual es un estilo de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es

decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

1.2.8 Herramienta CASE de planificación.

Visual Paradigm.

Es una herramienta CASE (Ingeniería de Software Asistida por Computación). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal. Existe una alternativa libre y gratuita de este software, la versión Visual Paradigm UML 6.4 CommunityEdition (CommunityEdition, ya que existe la Enterprise, Professional, etc.).

Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos. Entre sus características se encuentran:

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Generación de código para Java y exportación como HTML.

Esta herramienta permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y mantenimiento del software. Aumenta el conocimiento informático de una empresa ayudando así a la búsqueda de soluciones para los requisitos. También permite la reutilización del software, portabilidad y estandarización de la

documentación, además del uso de las distintas metodologías propias de la Ingeniería de Software.

1.3. Descripción del proceso de negocio que será objeto de informatización.

El Órgano de Criminalística, se encarga de concurrir a los casos de su competencia y realizar los peritajes de menor complejidad, fundamentalmente peritajes diagnósticos de las disciplinas de biología criminalística y trazología, también se recogen los datos del proceso de trabajo pericial y los relacionados con los delitos y posteriormente se analizan en diferentes oficinas del centro.

La necesidad que existe en el Órgano de Criminalística, en cuanto al proceso gestión del trabajo pericial de los oficiales y peritos, deriva un conjunto de acciones destinadas a la implementación de un sistema informático, que permita el acceso a la información de las salidas operativas con facilidad, integrándole facilidad de consultar y manejar la base de datos dentro de sus privilegios.

El producto permitirá la gestión de los distintos tipos de información que se manejan en el órgano antes mencionado, lo cual está restringido por diferentes normas de seguridad, predefinidas por los clientes y los desarrolladores. En resumen, la propuesta se basa en el control de la información, mediante la realización de un sistema informático que permita acceder a la misma de una forma rápida, eficiente y segura.

1.3.1 Reglas del negocio.

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, puesto que regulan algún aspecto del negocio. El proceso de especificación implica que hay que identificarlas dentro del negocio, evaluar si son relevantes dentro del campo de acción que se está modelando e implementarlas en la propuesta de solución. Jacobson, Rumbaugh, y Booch, El Proceso Unificado de desarrollo de Software.

A continuación, se listan las **reglas del negocio** de mayor importancia:

1. Sólo los peritos gestionan el informe pericial.
2. Los peritos deben actualizar los datos cada vez que exista una salida operativa.
3. Todo informe pericial debe ser entregado a la Oficina de Control de Documentos Periciales.
4. El periodo de entrega de un informe es de 15 días, a partir de la llegada a la Oficina de Control de Documentos Periciales.
5. Después de la 5:00pm no se puede recoger muestras periciales hasta las 8:00am.

6. Sólo la Oficina de Control de Documentos Periciales es la encargada de asignar tareas a cada una de las especialidades del laboratorio.

Mejoras al proceso gestión del trabajo operativo y pericial.

- Informatizar la gestión del trabajo operativo y pericial de la criminalística con la integración de los datos de las salidas operativas.
- Reducir el tiempo de respuesta a la solicitud de información realizada por los directivos.
- Garantizar la agilidad y fiabilidad de los datos registrados a la hora de confeccionar el informe pericial.
- Mantener la información organizada y actualizada, haciendo más fácil la búsqueda.

Conclusiones parciales.

Después de haber realizado el análisis de los procesos y un estudio de los fundamentos teóricos de las tecnologías, se llegó a las siguientes conclusiones:

1. Como resultado de la caracterización del proceso de trabajo operativo y pericial, se llegó a la conclusión de que el mismo requiere informatización.
2. Fueron analizadas las tecnologías y herramientas utilizadas por política de la institución definidos por los desarrolladores y el usuario.
3. La utilización de la metodología SCRUM facilitará de forma más precisa el desarrollo del proceso, identificando los roles principales y auxiliares de SCRUM.

CAPITULO 2: DESARROLLO DEL SISTEMA INFORMÁTICO PARA LA GESTIÓN DEL TRABAJO OPERATIVO Y PERICIAL EN LA CRIMINALISTICA.

Para darle solución a la problemática planteada, en este capítulo, se describe el sistema informático propuesto para la gestión del trabajo operativo y pericial del Órgano de Criminalística del MININT en la provincia Guantánamo, se especifican los requerimientos funcionales y no funcionales del sistema, se presentan los flujos de análisis y diseño junto a sus diagramas, se muestra el diagrama de despliegue y el modelo de datos, además del diseño de casos de prueba y un estudio de factibilidad donde se realiza una estimación de tiempo y esfuerzo basada en casos de uso.

2.1 Especificación de los requisitos de software.

Un requisito de software es una condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, un estándar u otro documento impuesto formalmente, que le permita al usuario resolver un problema o lograr un objetivo. Pressman, Ingeniería del Software, Un enfoque práctico 1.

2.1.1 Requerimientos funcionales.

Un requerimiento funcional, es un requisito que especifica una acción que debe ser capaz de realizar el sistema, sin considerar restricciones físicas; requisito que especifica comportamiento de entrada/salida de un sistema.

Para la herramienta propuesta se determinaron los siguientes **requisitos funcionales**:

R.F 1-Gestionar usuario.

R.F 1.1- Insertar usuario.

R.F 1.2- Modificar usuario.

R.F 1.3- Eliminar usuario.

R.F 1.4- Roles de usuario.

R.F 1.5- Mostrar trazas del sistema.

R.F 2- Crear planilla del trabajo operativo del perito en el lugar del suceso.

R.F 2.1- Mostrar planilla del trabajo operativo.

R.F 2.2- Enviar planilla a la (OCDP).

R.F 2.3- Exportar planilla.

R.F 3- Mostrar listado de planillas recibidas.

R.F 3.1-Mostrar planillas recibidas.

R.F 3.2-Mostrar planillas enviadas.

R.F 3.3- Enviar al laboratorio.

R.F 4- Mostrar listado de salidas operativas.

R.F 4.1-Agregar salida operativa.

R.F 4.2-Eliminar salida operativa.

R.F 4.3-Editar salida operativa.

R.F 5- Gestionar datos.

2.1.2 Requerimientos no funcionales.

Un requerimiento no funcional es una propiedad o cualidad que el producto debe tener, estos deben pensarse como las características que hacen al producto atractivo, ya que en muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Jacobson

RNF1-Apariencia o interfaz externa: Describe la apariencia del producto, cómo se pretende que sea la interfaz externa del producto o la necesidad de cumplir con normas, estándares, las interfaces de sistemas o los estándares de la empresa. El sistema brinda una interfaz sencilla, simple y fácil de usar permitiéndole además a cada usuario personalizar la visualización de la aplicación escogiendo entre más de diez temas disponibles como son Bootstrap, Material y Metro. Está diseñado de modo tal que el usuario pueda navegar entre las distintas acciones del sistema con gran facilidad.

RNF2-Usabilidad: Describen los niveles apropiados de uso. Derivan de una combinación de lo que el cliente está tratando de lograr con el producto y lo que los usuarios finales esperan del mismo. El sistema garantizará un acceso fácil y rápido a los usuarios para que no tengan dificultad al interactuar con el mismo, podrá ser usado por cualquier persona con conocimientos básicos sobre el manejo de la computadora y un ambiente web en sentido general.

RNF3-Rendimiento: Imponen condiciones a los requisitos funcionales tales como: velocidad de procesamiento o cálculo, eficiencia, disponibilidad, precisión, tiempo de respuesta o de recuperación. El sistema deberá ser capaz de procesar con rapidez y eficiencia los datos, con tiempos de respuesta y recuperación mínimos. Esto se garantiza gracias a que para el desarrollo del sistema fueron empleados marcos de trabajo como Bootstrap para el lado del servidor y jQuery para el lado del cliente que permiten desarrollar aplicaciones web de gran rendimiento y no sobrecargan el servidor.

RNF4-Portabilidad: el producto podrá correr en cualquier plataforma que sea capaz de levantar un navegador web que implemente los estándares de las diferentes tecnologías de entorno al desarrollo web.

RNF5 - Soporte: para trabajar con el sistema se requiere de un entorno de desarrollo Web funcionando: el servidor Web (Apache Server versión 2.0), el gestor de bases de datos (Oracle 11g), y PHP 5.8.2. Para el lado del cliente un navegador, como IE7, Firefox 8 o versiones superiores de éstos, Además se impartirá una preparación a los usuarios finales con la explicación de cómo se realiza el trabajo con el software.

RNF6-Seguridad: Al sistema se accederá a través de la autenticación: usuario y contraseña. La contraseña será evaluada dando un nivel de fortaleza de la misma para el usuario. La contraseña viajara por la red de forma encriptado desde el cliente hasta el servidor de base de datos. Se deben registrar además las trazas con el objetivo de conocer en cualquier instante las acciones realizadas por los distintos usuarios autenticados. Se garantizará que la información solo pueda ser vista por los usuarios con el nivel de acceso adecuado, además las funcionalidades del sistema se mostrarán de acuerdo al rol del usuario que este activo, garantizándose también protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

RNF7-Software

- **PC cliente:** Independientemente del sistema operativo, debe contar con uno de los siguientes navegadores web: Mozilla Firefox 10 o superior, Opera o Google Chrome.
- **El servidor:** Independientemente del sistema operativo, debe contar con Apache Server 2.0 como servidor web, PHP 5.1.6 o superior como lenguaje del lado del servidor y Oracle 11g como gestor de base de datos.

RNF8-Hardware

- **PC Cliente (requisitos mínimos):** El sistema podrá ser operado en máquinas clientes con procesador Pentium III a 1.5 GHz de velocidad, 256 MB de memoria RAM y tarjeta de red Ethernet de 10/100 MB.
- **PC Servidor (requisitos mínimos):** Pentium 4, con 512 MB de RAM, 1 GHz de velocidad y 20 GB de disco duro como mínimo.

2.2 Descripción del Sistema.

2.2.1 Historias de usuarios para los casos críticos.

Las Historias de Usuario son técnicas utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. (Canós, Penadés, & Letelier, 2012) Las Historias de Usuario para los casos críticos de la presente investigación son las siguientes:

Historia de Usuario	
Número: 1	Usuario: Perito
Nombre historia: Crear planilla de trabajo operativo del perito en el lugar del suceso	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Biolkis Quintero Delgado	
Descripción: El usuario insertará plantilla del trabajo operativo del perito en el lugar del suceso.	
Validación: El usuario podrá insertar plantilla del trabajo operativo del perito en el lugar del suceso.	

Tabla 1: Crear Planilla del trabajo operativo del perito en el lugar del suceso.

Fuente: Elaboración propia

Historia de Usuario	
Número: 2	Usuario: Perito
Nombre historia: Enviar planilla a la (OCDP)	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 1

Programador responsable: Biolkis Quintero Delgado
Descripción: Quiero enviar la planilla con los datos recogidos en el lugar del hecho a la (OCDP)
Validación: El usuario puede enviar la planilla con los datos de la salida operativa y el listado de los peritos que han participado

Tabla 2: Enviar planilla a la (OCDP).

Fuente: Elaboración propia

Historia de Usuario	
Número: 3	Usuario: Perito
Nombre historia: Exportar Planilla	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Biolkis Quintero Delgado	
Descripción: Quiero exportar planilla con los datos recogidos en la salida operativa.	
Validación: El usuario puede exportar planilla,	

Tabla 3: Exportar planilla.

Fuente: Elaboración propia

2.2.2 Sprint Backlog.

La pila de sprint o sprint backlog, es la lista que descompone las funcionalidades del producto en las tareas necesarias para construir un incremento: una parte completa y operativa del producto. Es útil porque descompone el proyecto en tareas de tamaño adecuado para determinar el avance a diario; e identificar riesgos y problemas sin necesidad de procesos complejos de gestión. (Flexibilidad con Scrum. Juan Palacio. Edición Octubre-2008).

A continuación, se muestran las tareas desarrolladas durante las diferentes iteraciones del proyecto:

Sprint 1	Pila de Sprint: Gestionar Roles de Usuario, Gestionar Usuarios, Filtrar trazas, Autenticar usuario.
-----------------	--

Inicio:		Duración:20 días	
Tareas Pendientes: 0		Horas de Trabajo Pendientes: 0	
Backlog ID	Tareas	Estado	Responsable
1	Crear una interfaz para Autenticar usuario	Terminado	Biolkis Quintero
2	Crear una interfaz para gestionar los roles de usuario	Terminado	Biolkis Quintero
3	Crear una interfaz para gestionar Usuario (Insertar, Modificar, Desactivar o Activar Usuario, cambiar contraseña.)	Terminado	Biolkis Quintero

Tabla 4: Gestionar Roles de Usuario, Gestionar Usuarios, Autenticar usuario

Fuente: Elaboración propia

Sprint 2		Pila de Sprint: Exportar (informe pericial, informe operativo).	
Inicio:		Duración:22 días	
Tareas Pendientes: 0		Horas de Trabajo Pendientes: 0	
BacklogID	Tareas	Estado	Responsable
1	Crear nuevo acceso en el menú para: Crear planilla del trabajo operativo del perito en el lugar del suceso	Terminado	Biolkis Quintero
2	Crear interfaz usuario donde se seleccione el rango de fecha deseado para obtener el informe.	Terminado	Biolkis Quintero

3	Crear una página controladora que procese los datos recibidos de la interfaz y llame al método correspondiente.	Terminado	Biolkis Quintero
4	Crear procedimientos almacenados en la base de datos que obtengan las salidas operativas seleccionadas según el rango de fecha insertada y la existencia final de las mismas según los datos recibidos de la página controladora.	Terminado	Biolkis Quintero
5	Crear método para exportar desde la interfaz el documento con los resultados obtenidos.	Terminado	Biolkis Quintero

Tabla 5: Sprint Backlog: Mostrar información de las salidas operativas.

Fuente: Elaboración propia

2.3 Análisis del sistema.

El objetivo del Análisis es traducir los requisitos a una especificación que describe cómo implementar el sistema. El análisis consiste en obtener una visión del sistema, de modo que sólo se interesa por los requisitos funcionales. Durante este flujo analizaremos los requisitos que se describieron anteriormente, refinándolos y estructurándolos; el objetivo es conseguir una comprensión más precisa de estos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar todo el sistema incluyendo su arquitectura.

2.3.1 Diagrama de comunicación.

A continuación, se muestran los diagramas de comunicación para los casos críticos:

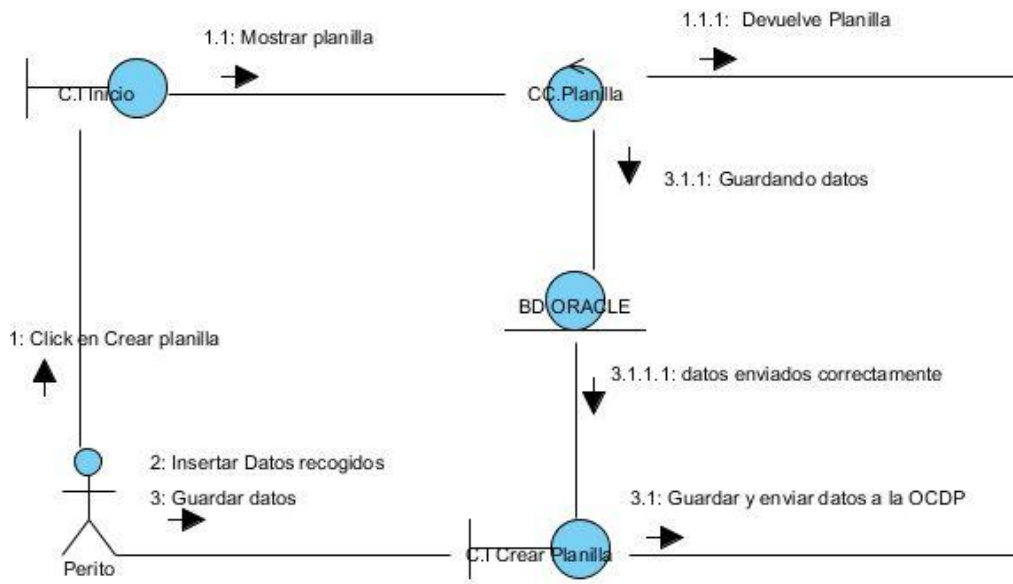


Figura 3. Diagrama de comunicación para: Crear planilla del trabajo operativo del perito en el lugar del suceso.

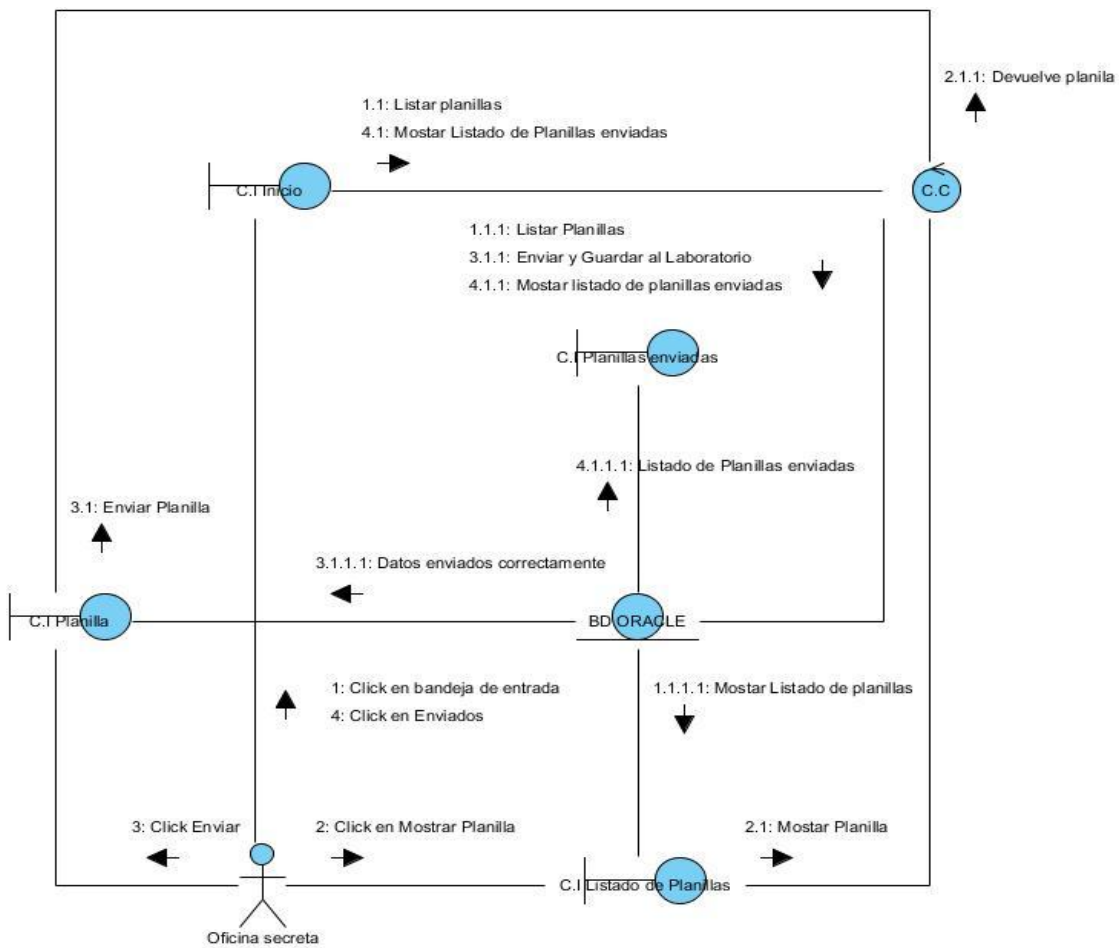


Figura 4: Diagrama de comunicación para la Oficina de Control de Documentos Periciales.

2.4 Diseño del Sistema.

En el diseño se modela el sistema en construcción de forma que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Además, se define una estructura del sistema que se debe conservar lo más fielmente posible durante el ciclo de vida completo del software.

El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema. El análisis consiste en obtener una visión del sistema, de modo que sólo se interesa por los requisitos funcionales. Por otro lado, el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales.

2.4.1 Patrones de diseño.

Un patrón de diseño describe una estructura de diseño que resuelve un problema particular del diseño dentro de un contexto específico y entre “fuerzas” que afectan la manera en la que se aplica y en la que se utiliza dicho patrón.

El objetivo de cada patrón de diseño es proporcionar una descripción que permita a un diseñador determinar ,1) si el patrón es aplicable al trabajo en cuestión, 2) si puede volverse a usar (con lo que se ahorra tiempo de diseño) y 3) si sirve como guía para desarrollar un patrón distinto en funciones o estructura.

Los patrones de diseño son la base para la búsqueda de soluciones a los problemas comunes durante el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces, contribuyendo a dar flexibilidad y extensibilidad al diseño. Uno de los patrones más usados son los patrones GRASP, los mismos son patrones generales de software para asignación de responsabilidades que tienen como propósito general asignar responsabilidades expresadas en forma de patrones.

También podemos encontrar el patrón de arquitectura de software Modelo Vista Controlador (MVC), a continuación, se menciona el patrón más relevante utilizado en el módulo informático propuesto:

El patrón MVC realiza un diseño que separa los datos y la lógica de negocio, con el objetivo de mejorar la reusabilidad, permitiendo esto que las modificaciones realizadas en las vistas influyan en menor medida en la lógica de negocio o de datos.

El propósito del MVC es aislar los cambios. Es una arquitectura preparada para los cambios, que desacopla datos y lógica de negocio de la lógica de presentación, permitiendo la actualización y desarrollo independiente de cada uno de los citados componentes.

El MVC consta de:

Una o más vistas de datos.

Un modelo, el cual representa los datos y su comportamiento.

Un controlador que controla la transición entre el procesamiento de los datos y su visualización.

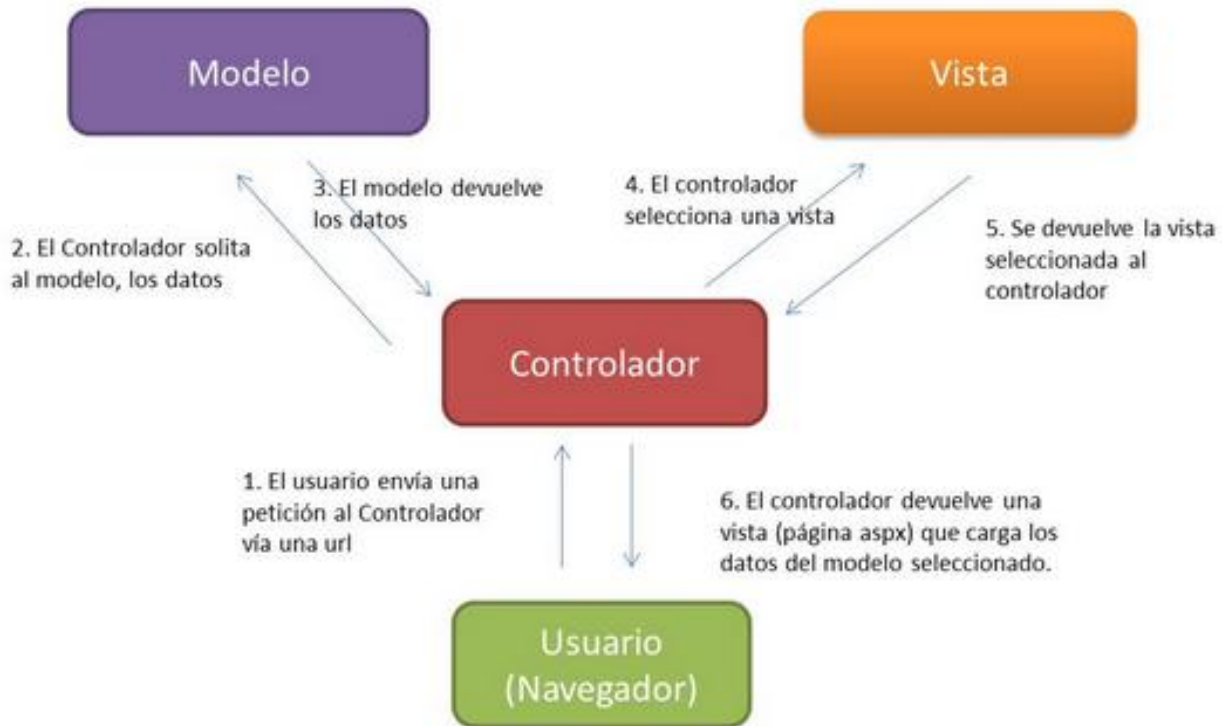


Figura 5. Patrón de diseño MVC.

2.4.2 Descripción de la arquitectura del software.

Arquitectura es la estructura de los componentes más significativos de un sistema interactuando a través de interfaces con otros componentes conformados por componentes sucesivamente pequeños e interfaces. (Jacobson, Rumbaugh, &Booch, 2006)

Arquitectura Cliente-Servidor: Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados "servidores". La arquitectura cliente-servidor puede ser clasificada en dos capas, tres capas o n capas.

A continuación, se muestran algunas ventajas de la arquitectura cliente-servidor:

Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.

Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado.

Los cambios realizados en las plataformas de los clientes o de los servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.

Las funciones de cliente y servidor pueden estar en plataformas separadas, o en la misma plataforma.

Un servidor da servicio a múltiples clientes en forma concurrente.

2.4.3 Modelo de Entidad Relación.

Para la obtención del diagrama entidad relación se desarrolló un proceso de normalización donde las relaciones fueron llevadas hasta la 3ra forma normal, obteniéndose como resultado el modelo siguiente:

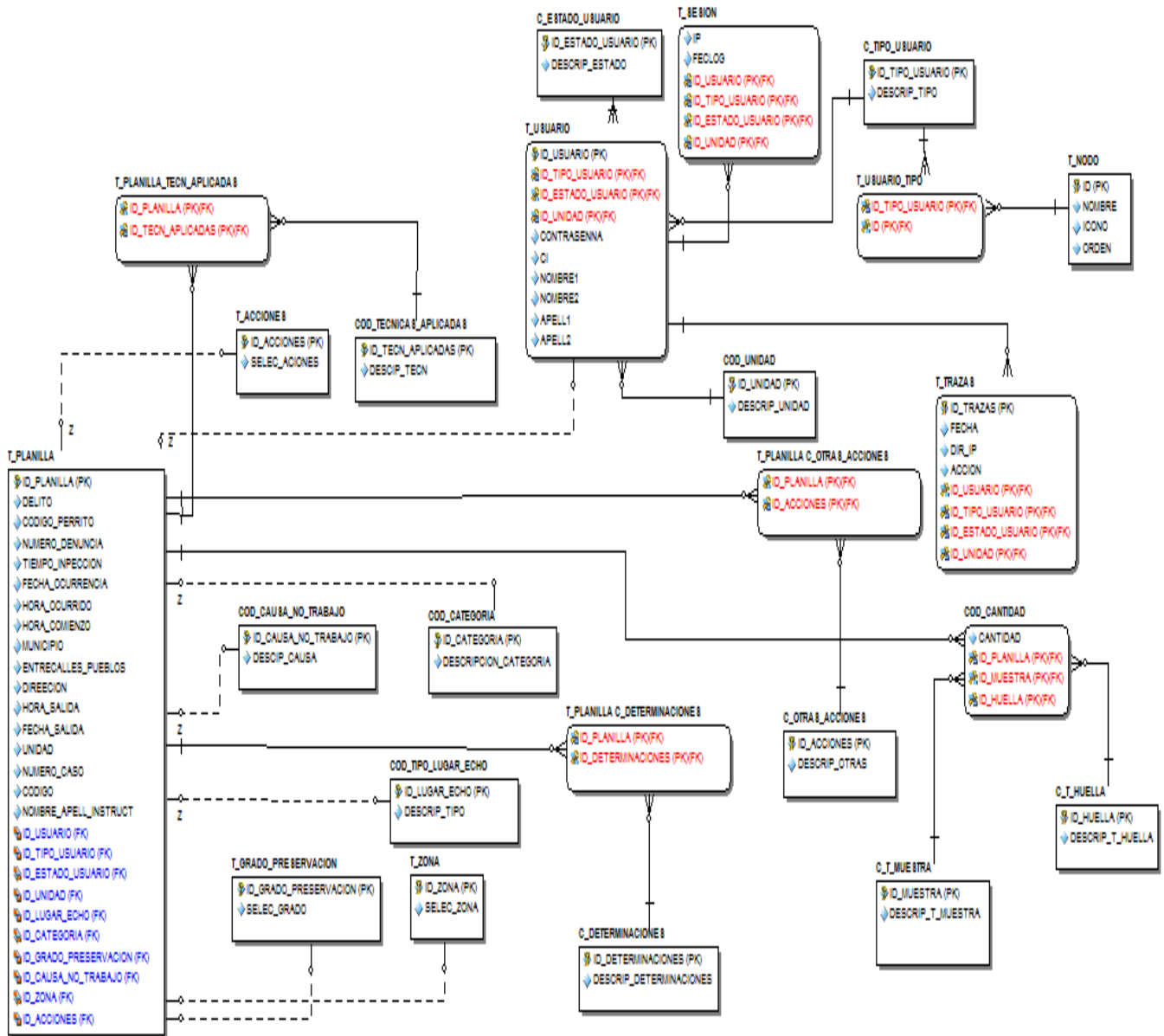


Figura 6: Modelo de Entidad Relación.

2.5 Implementación.

2.5.1 Diagrama de despliegue.

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).

Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución.

La distribución física del sistema propuesto es el siguiente:

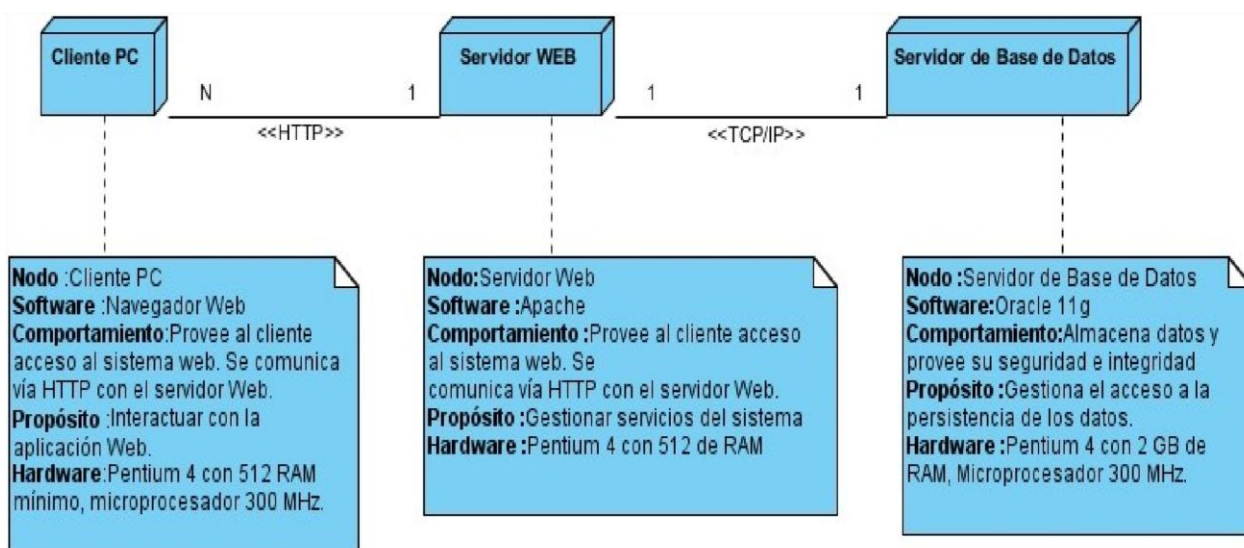


Figura 7: Diagrama de despliegue.

Nodo	Servidor de Base de Datos
Software	Oracle 11g
Comportamiento	Almacena datos y provee su seguridad e integridad.
Propósito	Gestiona el acceso a la persistencia de los datos.
Hardware	Pentium 4 con 2 GB de RAM, Microprocesador 300 MHz.

Tabla 6: Servidor de Base de Datos

Fuente: Elaboración propia.

Nodo	Cliente PC
Software	Navegador Web
Comportamiento	Provee al cliente acceso al sistema Web. Se comunica vía HTTP con el servidor Web.
Propósito	Interactuar con la aplicación Web.

Hardware	Pentium 4 con 512 RAM mínimo, microprocesador 300 MHz.
-----------------	--

Tabla 7: Cliente PC

Fuente: Elaboración propia.

Nodo	Servidor Web
Software	Apache
Comportamiento	Provee al cliente acceso al sistema Web. Se comunica vía HTTP con el servidor Web.
Propósito	Gestionar servicios del sistema.
Hardware	Pentium 4 con 512 de RAM.

Tabla 8: Servidor Web.

Fuente: Elaboración propia.

2.6 Diseño de caso de prueba.

Los casos de prueba son un conjunto de condiciones o variables bajo las cuáles el analista determina si el requisito de una aplicación es parcial o completamente satisfactorio. Se pueden realizar muchos casos de prueba para determinar que un requisito es completamente satisfactorio. Con el propósito de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba para cada requisito a menos que un requisito tenga requisitos secundarios. En ese caso, cada requisito secundario deberá tener por lo menos un caso de prueba.

Uno de ellos debe realizar la prueba positiva de los requisitos y el otro debe realizar la prueba negativa. El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada.


Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica.

2.6.1 Prototipos de la interfaz de usuario para los casos críticos.

Los prototipos de la interfaz de usuario son la presentación de la interfaz del producto que representa la funcionalidad contenida en los casos de uso; de manera que permita que el usuario verifique que el sistema va a satisfacer sus necesidades. Se utilizan para dar al usuario una vista preliminar de parte del software. La construcción de estos contribuye a que el software sea de mejor calidad, además de que reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios.

2.6.2 Prueba de caja negra.

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software; los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Caso de uso del sistema	Autenticar usuario
Caso de prueba	Comprobar que no se puede acceder al sistema con usuario o contraseña incorrecta.
Imagen	
Descripción	Se intentará acceder al sistema con un usuario válido y una contraseña incorrecta.
Entrada	Usuario: BIOLKIS Contraseña: *****


Condiciones de Ejecución	El usuario BIOLKIS no está registrado en el sistema.
Resultados esperados	

Tabla 9: Diseño de prueba de caja negra del caso de prueba: Comprobar que no se acceda al sistema con un usuario y una contraseña incorrecta.

Fuente: Elaboración propia

Prototipos

Mostrar Planillas Recibidas, Enviadas y listado de planillas

Bienvenido, ocdp



Criminalística Gtmo

Inicio 🏠

Bandeja de Entrada 📧

Enviados 📧

Mostrar Planilla

Listado de Planillas recibidas

Seleccionar	Nombre y apellido del perito	Delito	Codigo	No Caso	Nombre y apellido del instructor	Unidad
📄						

Tabla 10: Mostrar Salidas Operativas.

Fuente: Elaboración propia

Prueba de aceptación

Pruebas de Aceptación (prueba alfa y prueba beta).

Las pruebas de aceptación, también llamadas pruebas del cliente, son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptación se derivan de las historias de los usuarios que se han implementado como parte de la liberación del software (Pressman, R.S., Ingeniería del Software un enfoque práctico 2010).

La prueba alfa se lleva a cabo en el sitio del desarrollador por un grupo representativo de usuarios finales. El software se usa en un escenario natural con el desarrollador "Observando" a los usuarios y registrando los errores y problemas de uso. Las pruebas alfa se realizan en un ambiente controlado

La prueba beta se realiza en uno o más sitios del usuario final. A diferencia de la prueba alfa, por lo general el desarrollador no está presente. Por tanto, la prueba beta es una aplicación "en vivo" del software en un ambiente que no puede controlar el desarrollador. El cliente registra todos los problemas (reales o imaginarios) que se encuentran durante la prueba beta y los reporta al desarrollador periódicamente. Como resultado de los problemas reportados durante las pruebas beta, es posible hacer modificaciones y luego preparar la liberación del producto de software a toda la base de clientes. (Pressman, R.S., Ingeniería del Software un enfoque práctico 2010).

2.7 Estudio de factibilidad.

Una vez establecidos los casos de uso que guiarán el desarrollo del software, se puede predecir una estimación del tiempo de duración del proyecto mediante el análisis de puntos de Casos de Uso. La estimación mediante el análisis de Puntos de Casos de Uso se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. (Peralta, Mario. Estimación del esfuerzo basado en casos de uso. Buenos Aires, Argentina: Centro de Ingeniería del Software e Ingeniería del Conocimiento (CAPIS). 2007.) Determinar el costo de elaboración del proyecto

que se desarrolla, los beneficios que aporta para sus usuarios y la verdadera necesidad e importancia de que la aplicación se implante.

2.7.1 Planificación basada en casos de uso.

Paso 1. Cálculo de Puntos de Casos de Uso sin ajustar (PCU)

Factor de Peso de Actores (FPA): Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos.

Actores	Peso	Tipo
Administrador del Sistema	3	Complejo
Perito o Instructor	3	Complejo
Servicio de la oficina secreta	2	Medio
Servicio del Laboratorio	2	Medio
FPA = 3 x 3 + 2 x 2 = 13		

Tabla 11: Factor de peso de los actores.

Fuente: Elaboración propia

Factor de Peso de los Casos de Uso sin ajustar (FPCU): Este valor se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos.

Casos de Uso	Complejidad	Peso correspondiente
Crear Planillas	Complejo	15
Gestionar Usuarios	Complejo	15
Gestionar Consultas	Complejo	15
Exportar informe	Medio	10
FPCU = 15x3 + 10 = 40		

Tabla 12: Factor de peso de los casos de uso

Fuente: Elaboración propia

Con los datos de FPA y FPCU obtenidos se puede determinar que:

$$PCU = FPA + FPCU$$

$$PCU = 13 + 40$$

$$PCU = 53$$

Paso 2. Calcular los Puntos de Casos de Uso Ajustados (PCUA)

Ya obtenidos los Puntos de Casos de Uso sin ajustar, se debe ajustar este valor mediante la ecuación:

$$PCUA = PCU * FCT * FA \text{ donde:}$$

PCU: Puntos de Casos de Uso.

PCUA: Puntos de Casos de Uso Ajustados.

FCT: Factor de Complejidad Técnica.

FA: El Factor de Ambiente.

Calcular el Factor de Complejidad Técnica (FCT)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestran los factores tenidos en cuenta y el peso de cada uno de estos:

Calcular el Factor de Complejidad Técnica (FCT)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestran los factores tenidos en cuenta y el peso de cada uno de estos:

Factor	Descripción	Peso	Criterio	Valor
1	Sistema distribuido.	2	El sistema es centralizado.	5

2	Objetivos de performance o tiempo de respuesta.	1	El sistema responde rápidamente a las solicitudes de los usuarios.	5
3	Eficiencia del usuario final.	1	Los usuarios finales deben poseer conocimiento en el trabajo con aplicaciones web.	5
4	Procesamiento interno complejo.	1	El procesamiento interno del sistema tiene una complejidad media.	5
5	El código debe ser reutilizable.	1	El código del sistema es reutilizable.	5
6	Facilidad de instalación.	0.5	El sistema es fácil de instalar.	5
7	Facilidad de uso.	0.5	El sistema es fácil de usar.	5
8	Portabilidad.	2	El sistema es multiplataforma.	5
9	Facilidad de cambio.	1	El sistema puede ser cambiado con facilidad.	5
10	Concurrencia.	1	Pueden existir varios usuarios trabajando en el sistema al mismo tiempo.	5
11	Incluye objetivos especiales de seguridad.	1	Para la realización de acciones en el sistema se deben establecer mecanismos especiales de seguridad.	5
12	Provee acceso directo a terceras partes.	1	No	0

13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	Se requiere capacitar al personal que va a trabajar con la aplicación.	5
$FCT = 0.6 + 0.01 * \sum(\text{Peso } i * \text{Valor } i)$ $FCT = 1.3$				

Tabla 13: Factor de complejidad técnica.

Fuente: Elaboración propia

Calcular el Factor de Ambiente (FA)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del FA. El cálculo del mismo es similar al cálculo del FCT, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Haciendo un análisis de lo que plantea cada factor a medir, se asignaron los siguientes valores a cada factor:

Factor	Descripción	Peso	Valor
1	Familiaridad con el modelo de proyecto utilizado	1.5	5
2	Experiencia en la aplicación	0.5	4
3	Experiencia en orientación a objetos	1	3
4	Capacidad del analista líder	0.5	5
5	Motivación	1	5
6	Estabilidad de los requerimientos	2	5
7	Personal a tiempo parcial	-1	0
8	Dificultad del lenguaje de programación	-1	4

$$FA = 1.4 + 0.03 \times \sum (\text{Peso}_i * \text{Valor}_i)$$

$$FA = 2.18$$

Tabla 14: Factor Ambiente.

Fuente: Elaboración propia

En estos momentos se cuenta con los valores de PCU, FCT y FA, por tanto, se puede calcular el valor de PCUA:

$$PCUA = PCU * FCT * FA$$

$$PCUA = 99 * 1.3 * 2.18$$

$$PCUA = 280.566$$

Paso 3. Estimación de esfuerzo a través de los puntos de casos de uso (E)

$$E = PCUA \times FC \text{ donde:}$$

E: Esfuerzo estimado en horas hombres.

FC: Factor de Conversión = 20 H/H.

$$E = 280.566 * 20 = 5611.32 \text{ Horas-Hombre para implementación.}$$

Paso 4. Cálculo del esfuerzo de todo el proyecto E (Total).

Este esfuerzo es el que se requiere para la implementación. Si se tiene en cuenta que este representa un 50% del esfuerzo total para desarrollar el software entonces tenemos que el esfuerzo total es el siguiente:

$$E (\text{Total}) = E / 0.5$$

$$E (\text{Total}) = 5611.32 / 0.5$$

$$E (\text{Total}) = 11222.64 \text{ Horas-Hombre}$$

Actividad	Porcentaje	E (H/H)
Análisis	10	1122.264
Diseño	20	2244.528
Implementación	50	5611.32

Pruebas	10	1122.264
Sobrecarga (otras actividades)	10	1122.264
Total	100	11222.64

Tabla 15: Esfuerzo por cada actividad del proyecto.

Fuente: Elaboración propia

Estimación del tiempo de desarrollo del proyecto (TDES)

$$\text{TDES} = 11222.64 \text{ HH} / 2 \text{ Hombre}$$

$$\text{TDES} = 5611.32 \text{ Horas.}$$

Paso 5. Calcular estimación del costo de desarrollo del proyecto C (total).

Una vez estimado el tiempo de desarrollo del proyecto y conociendo la cantidad de desarrolladores y el pago que recibe cada uno de estos se puede llevar a cabo una estimación del costo total del proyecto referidos a los recursos humanos; existen otros costos como por ejemplo del equipamiento que se suman al anterior. El costo por concepto de desarrolladores viene dado por:

$$C(\text{total}) = E(\text{Total en HH}) * \text{CHH}$$

CHH: Costo por hombre hora

$$\text{CHH} = K * \text{THP}$$

Donde:

K: Coeficiente que tiene en cuenta los costos indirectos (1.5 y 2.0).

THP: Tarifa Horaria Promedio. El salario promedio de las personas que trabajan en el proyecto dividido entre 160 horas.

El salario promedio del desarrollador es de \$100 y,

$$\text{por tanto: THP} = 100 / 160 = 0.625$$

Entonces:

$$C(\text{total}) = E(\text{Total}) * K * \text{THP}$$

$$C(\text{total}) = 11222.64 * 2 * 0.625$$

C (total) = \$ 14028.3

Análisis económico.

Una vez estimado el sistema, se procede al análisis económico para evaluar la factibilidad del desarrollo del sistema. Se hace necesario identificar los costos tangibles e intangibles más los beneficios tangibles e intangibles relacionados con la implementación de la propuesta:

Costos Tangibles

Gastos en papel.

Gastos en Tóner.

Gastos de llamada telefónicas.

Costos Intangibles

Posible resistencia al cambio.

Cambios en la forma de trabajo.

Beneficios Tangibles

Ahorro de horas- hombres.

Ahorro en Tóner.

Ahorro en llamadas telefónicas.

Ahorro en paquetes de hojas.

Beneficios Intangibles

Mejoras en la integridad, confiabilidad y calidad de la información para una mejor gestión.

Ahorro de tiempo en la manipulación de la información.

Mejoras en cuanto al control y seguridad de la información.

Asegura la continua participación y colaboración de todo el personal en el proceso.

Disponibilidad de mecanismos para una mejor gestión y optimización de procesos.

Humanización del trabajo.

Beneficios	Gastos antes(A)	Gastos después (B)	UM	Precio Ahorro unitario (C)	(A-B)*C
Ahorro en Horas-Hombres	1765	182	Horas	\$3,61	\$5714,63
Ahorro en Tóner	20	7	Unidad	\$32.00	\$416
Ahorro en Hojas	60	20	Paquete	\$42,47	\$1698,8
Ahorro en llamadas telefónicas	15500	5000	Minuto	\$0,20	\$2100
Ahorro Total Anual					\$9929,43

Tabla 16: Cuantificación de los beneficios tangibles.

Fuente: Elaboración propia

Punto de Equilibrio: El punto de equilibrio, en términos de contabilidad de costos, es aquel punto de actividad en donde los ingresos son iguales a los costos, es decir, es el punto de actividad en donde no existe utilidad ni pérdida. Para tener una visión general de los gastos actuales y futuros en caso del establecimiento del sistema se describen a continuación un antes y un después del mismo.

Recursos	Costo actual	Costo del sistema propuesto
Horas-Hombres	\$6371,65	\$657,02
Tóner	\$640	\$224
Hojas	\$2548,2	\$849,4
Llamadas telefónicas	\$3100	\$1000

Total anual	\$12659,85	\$2730,42
Costo del sistema	\$0	\$9929,43

Tabla 17: Relación de costos en los que se incurren antes y después del sistema.

Fuente: Elaboración propia

Después se define la variable discreta cantidad de órganos, esta al ser aplicada expone los costos en la forma actual y con el sistema propuesto.

Cantidad de Órganos	Costos del sistema actual	Costos del sistema propuesto
1	\$12659,85	\$12540,14
2	\$25319,7	\$15390,27

Tabla 18: Relación de costos teniendo en cuenta la cantidad de órganos.

Fuente: Elaboración propia.

Luego de analizados los costos, es posible obtener el punto de equilibrio el cual se muestra a continuación:

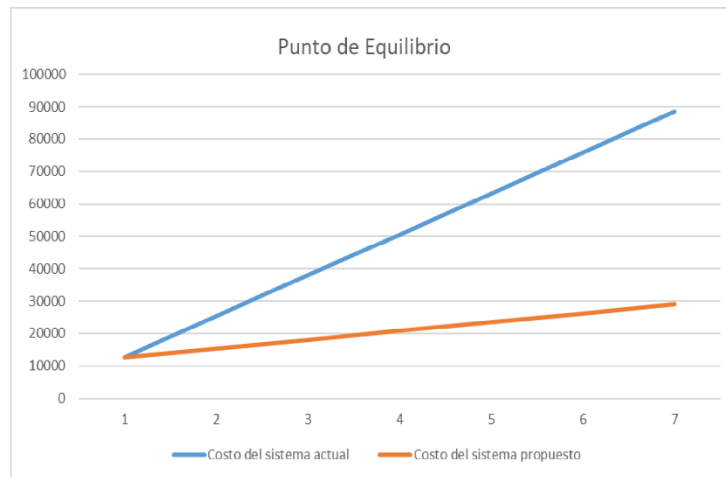


Figura 8: Punto de equilibrio.

Año	Costos	Costos Acumulados	Beneficios	Beneficios Acumulados	Flujo Efectivo	Flujo Acumulados
0	\$12659,85	\$12659,85			-\$12659,85	-\$12659,85
1	\$2730,42	\$15390,27	\$5209	\$5209	\$2478,58	-\$10181,27
2	\$2730,42	\$18120,69	\$7687,58	\$12896,58	\$4957,16	-\$5224,11
3	\$2730,42	\$20851,11	\$12644,74	\$25541,32	\$9914,32	\$4690,21
4	\$2730,42	\$23581,53	\$22559,06	\$48100,38	\$19828,64	\$24518,85
5	\$2730,42	\$26311,95	\$42387,7	\$90488,08	\$39657,28	\$64176,13
VAN		\$11.127,93		TIR		60%

Tabla 19: Retorno de la Inversión

Fuente: Elaboración propia

Retorno de la inversión:

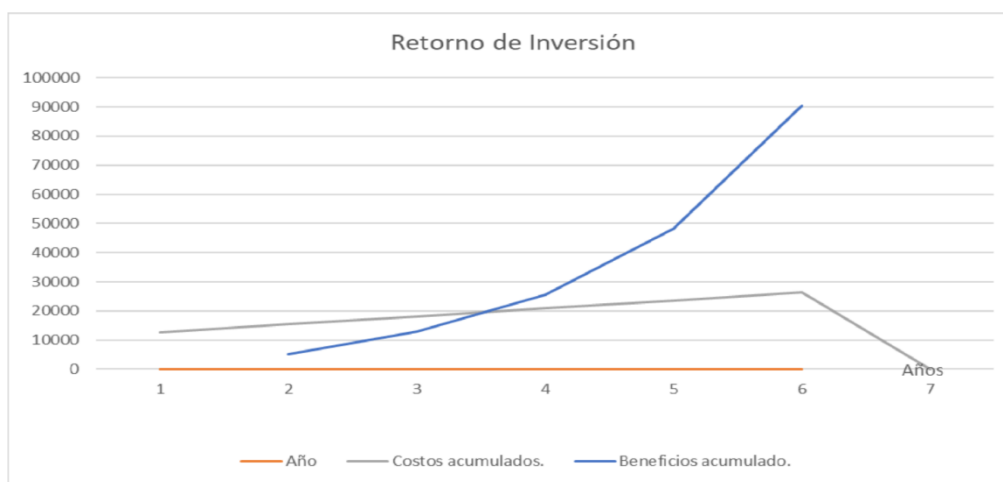


Figura 9: Retorno de la inversión.

Punto de Retorno de la Inversión (PRI) en años: El Punto de Retorno de la Inversión es el tiempo en que se igualan los costos y los beneficios acumulados; y es entonces, cuando se recupera la inversión y se comienza a obtener ganancias. $PRI = N-1 + \text{ABS} (FAN-1 / FN)$

Donde:

N: Año en que el flujo acumulado cambia de signo

FAN-1: Flujo de efectivo acumulado en el año previo a "N"

FN: Flujo neto de efectivo en el año N

$PRI = 2 - 1 + ABS(-11304,84 - 1/32000,52)$

$PRI = 1 + ABS(-11304,84 / -32000,52)$

$PRI = 1 + 0,353270$

$PRI = 2,5269256 \approx 1 \text{ año}$

El VAN (Valor Actual Neto) es mayor que cero y el TIR (Tasa Interna de Retorno) es mayor del 60%; por tanto, se puede afirmar que el sistema propuesto es factible.

Conclusiones parciales.

1. Se realizó la captura de requisitos funcionales y no funcionales que debe cumplir el sistema informático propuesto.
2. Después de aplicado el análisis de factibilidad quedó determinada la viabilidad de desarrollo del sistema informático propuesto.
3. Se realizó el análisis y el diseño; se diseñó la base de datos que contiene los datos persistentes del sistema; el diagrama de despliegue: que muestra la configuración física de los nodos y se realizaron pruebas de caja negra para detectar posibles errores.

CONCLUSIONES GENERALES.

1. Se caracterizó el proceso gestión del trabajo operativo y pericial en el Órgano de Criminalística del MININT en Guantánamo, identificándose las limitaciones y deficiencias, lo que demostró la necesidad de implementar un sistema informático que posibilite que este proceso se lleve a cabo de manera eficiente.
2. Se desarrolló un sistema informático que permite elevar la eficiencia en el proceso gestión del trabajo operativo y pericial en el Órgano de Criminalística del MININT en Guantánamo.

RECOMENDACIONES.

Luego de realizada la investigación y teniendo en cuenta los resultados obtenidos, se recomienda:

1-Generalizar el uso del sistema desarrollado en otras provincias del país.

BIBLIOGRAFÍA.

1. @Manz. (2018). Lenguaje CSS - Documentación sobre diseño web. <https://lenguajecss.com/>
2. Ault, M. (2017). Tuning Third-party Vendor Oracle Systems Tuning when you can't touch the code. ISBN: 0-9740716-3-3
3. Aley, R. (2016). PHP Beyond the Web
4. Aley, R. (2017). Pro Functional PHP Programming. Application Development Strategies for Performance Optimization, Concurrency, Testability, and Code Brevity
5. Administración de Bases de datos. (17 de agosto de 2017). Obtenido de Los Requisitos mínimos de instalación de Oracle 11G en Windows: <http://mory199.blogspot.com/2017/08/los-requisitosminimos-de-instalacion-oracle11g>
6. Álvarez., C., COMO SE MODELA LA INVESTIGACION CIENTIFICA. 1ra Edición. ed. Vol. 1ra Parte. 2013, Habana, Cuba. 91
7. Alarcón, R. (2000). Diseño orientado a objetos con UML. Madrid, España: Grupo EIDOS Consultoría y Documentación Informática, S.L.
8. Acedo, J. (04 de 05 de 2015). Web: ¿Qué es el Framework Bootstrap? Ventajas y Desventajas. <http://programacion.jias.es/2015/05/web-%C2%BFque-es-el-framework-bootstrap-ventajas-desventajas/>
9. ALEGSA.com.ar. (12 de 11 de 2010). Ventajas y desventajas del modelo cliente-servidor. <http://www.alegsa.com.ar/Respuesta/ventajas-y-desventajas-del-modelo-clienteservidor.htm>.
10. Bakken, S. S., Aulbach, A., Schmid, E., Winstead, J., Wilson, L. T., Lerdorf, R., . . . Ahto, J. (2001). Manual de PHP. (R. Martínez, Ed.) Grupo de documentación de PHP.
11. Bootstrap team. (2019). Bootstrap - The most popular HTML, CSS, and JS library in the world. Obtenido de <https://getbootstrap.com/>
12. Burleson, D., Hordila, M. (2017) Oracle Index Management Secrets Top Oracle Experts Discuss Index Management Techniques. ISBN: 0-9740716-9-2
13. Criminalística de Laboratorio, Dr. Romualdo Ayala Ponce, pág. 7
14. Jeff, C. (2019).
15. Definición. DE. (03 de 08 de 2017). Definición de HTML. <https://developer.mozilla.org/es/docs/Web/HTML>
16. Edmunds, B. (2016). Securing PHP Apps.

17. Foot, C. (2017) OCP Instructors Guide for Oracle DBA Certification A Study Guide to Advanced Oracle Certified Professional Database Administration Techniques. ISBN: 0-9740716-1-7
18. Galen, R. (2019). Scrum Product Ownership: Navigating The Forest AND The Trees. ISBN-13: 978-0988502642, ISBN-10: 098850264
19. Hurtado, G. S. V. (2009). "Representación de la arquitectura de software usando UML.
20. Haverbeke, M. (2018). EloquentJavaScript3rdedition.
21. Jacobson, B., &Rumbaugh, UML y Patrones.
22. Jacobson, I. B., Grady; Rumbaugh, James. . . (2004). El Proceso Unificado de Desarrollo de Software (Editorial Félix Varela. ed.). La Habana. Cuba.
23. Jacobson, I. B., Grady; Rumbaugh, James.,. (2006). El Proceso Unificado de Desarrollo de Software. (Editorial Félix Varela. ed. Vol. Volumen 1). La Habana. Cuba.
24. Jacobson, R., &Booch, (2006). El Proceso Unificado de desarrollo de Software
25. Kruchten, P. (2010). "Arquitectura de Software."
26. LucidChart. (14 de 05 de 2014). ¿Qué es un diagrama entidad-relación?
<https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>
27. Letelier, P. M. P., Carmen. ((2008)). "Metodologías ágiles para el desarrollo de software. Universidad Politécnica de Valencia.
28. Lengstorf, J. (2010). Pro PHP and jQuery.
29. Larman, C. (2003). UML y Patrones (Segunda ed.). EUA: Pretice Hall, Inc.
30. Matos García, R. M. (2004). Sistema de Base de Datos. La Habana: Félix Varela. Moreno, op.cit, págs. 344-345.
31. MK@Marketing& Web. (01 de 01 de 2019). 20 Lenguajes de Programación más usados en el 2018. Recuperado el 10 de 05 de 2019, de 20 Lenguajes de Programación más usados en el 2018: <https://www.marketingandweb.es/marketing/lenguajes-de-programacion-mas-usados/>.
32. Microsoft. (30 de 06 de 2017). C# Language Specification. Recuperado el 20 de 02 de 2019, de C# LanguageSpecification: <https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/language-specification/introduction>.
33. Matishak, D., & Fuller, M. (2010). Oracle Database 11g: Administration Workshop I (Vol. I). (R. Kumar, & D. Milne, Edits.) United States: Oracle and/or it affiliates.
34. Palacio., J. (2008). Flexibilidad con Scrum (Edición Octubre ed.).
35. Peralta, M. (2007). Estimación del esfuerzo basado en casos de uso
36. Pérez, J. E. File Transfer ProtocolPRESSMAN, R. (2005). "Ingeniería de software: Un enfoque práctico " Vol. Parte I y II.

37. P. Lubbers, B. Albers, F. Salim, P. Lubbers, B. Albers, and F. Salim, "Using the html5 web storage api," in Pro HTML5 Programming, pp. 213–241, Apress, 2010.10.1007/978-1-4302-2791-5 9.
38. Pressman, R. S. (2010). Ingeniería de Software Un enfoque práctico (Séptima Edición ed.). (P. R. Vázquez, Ed.) México: Mc Graw Hill Educación. Rakhimov. (2008). B.R.A.E.S., Oracle PL/SQL by Example
39. Resig, J... (2009). Secrets of the JavaScript Nija
40. Stoner., J. (2010). Administración y la Dirección Integrada de Proyectos.
41. Schwaber , Ken y Sutherland , Jeff, La guía de Scrum, 2011.
42. Sampieri, Fernández Collado, C., & Baptista Lucio, M. d. (2014). Metodología de la Investigación (6 ed., Vol. 6). México: McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V.
43. Sistemas. (06 de 05 de 2017). Definición de Arquitectura Software. <https://sistemas.com/arquitectura-software.php>.
44. Tecnologías, U. I. (20 de 01 de 2016). Bootstrap, Ventajas y Desventajas. <http://tecnologiaenvivo.com/bootstrap-ventajas-y-desventajas/>
45. UNIWEBSIDAD. (08 de 12 de 2017). Introducción a CSS. Recuperado el 25 de 02 de 2018, de Introducción a CSS: <https://uniwebsidad.com/libros/css/capitulo-1>.
46. Vilalta., J., UML Guía Visual Cómo crear formas de vida organizativa. 2014.
47. Verheyen, G. (2018) Scrum – A Pocket Guide - 2nd edition: A Smart Travel Companion 2nd Edition.
48. West, Adrian, Prettyman, Steve (2018). Practical PHP 7, MySQL 8, and MariaDB Website Databases A Simplified Approach to Developing Database-Driven Websites.
49. Zayas., C.M.A.d., En la escuela de la Vida. Didáctica. 1ra Edición. ed. Vol. 1er Volumen. 2015, Habana, Cuba. 159.
50. Zandstra, M. (20160). PHP Objects, Patterns, and Practice.

GLOSARIO DE TÉRMINOS

CAPIS: Centro de Ingeniería del Software e Ingeniería del Conocimiento

GRASP: son patrones generales de software para asignación de responsabilidades, es el acrónimo de "GRASP (object-oriented design General Responsibility Assignment Software Patterns)". Aunque se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software.

GTK+: Es un conjunto de Bibliotecas Multiplataforma para desarrollar Interfaces gráficas de usuario (GUI), principalmente para los Entornos gráficos GNOME, XFCE y ROX, aunque también se puede usar en el Escritorio de Windows, MacOS y otros.

Herramientas CASE: (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadoras). Son diversas Aplicaciones informáticas destinadas a aumentar la productividad en el Desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el diseño de proyectos, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, Compilación automática, documentación o detección de errores entre otras.

Odorología Criminalística: Huellas olorosas.

Trazología: Huellas de dientes, de calzados, de neumáticos, de frenaje, de instrumentos de fractura, de animales.

ANEXO

Anexo 1: Entrevista a Directivos del Órgano de Criminalística, Personal y Preparación.

Objetivo: Obtener información sobre el proceso gestión del trabajo operativo y pericial del Órgano de Criminalística del MININT en la provincia Guantánamo.

Aspectos a evaluar: Gestión de la información correspondiente al proceso gestión del trabajo operativo y pericial llevado a cabo en las salidas operativas del Órgano de Criminalística del MININT en la provincia Guantánamo.

Presentación:

Compañero(a) estamos interesados en conocer de qué forma se desarrolla actualmente el proceso gestión del trabajo operativo y pericial llevado a cabo en las salidas operativas de dichos trabajadores del Órgano de Criminalística del MININT de la provincia Guantánamo y el grado de complejidad de las acciones que con frecuencia se llevan a cabo durante el proceso. Le pedimos, que, dentro de sus posibilidades, nos aporten alguna información al respecto.

Cuestionario:

1. ¿Cómo se gestiona actualmente la información concerniente al proceso del trabajo operativo y pericial en una salida operativa?
2. ¿A su juicio, ¿cuáles son las principales limitaciones que se presentan actualmente en la ejecución de este proceso?
3. ¿Cuáles son las normativas, órdenes y reglamentos que rigen la ejecución del proceso antes mencionado?
4. ¿A su juicio, sería factible para la institución contar con una herramienta que permita gestionar la información correspondiente al proceso del trabajo operativo y pericial?
5. ¿Puede sugerir algún aspecto para ser tenido en cuenta en la estructura del sistema?
6. ¿Actualmente con qué herramienta se registra el proceso del trabajo operativo pericial?

Resumen de los resultados de la entrevista:

Fue entrevistada una muestra de 4 directivos del Órgano de Criminalística del MININT de la provincia Guantánamo.

En la pregunta uno el total de los entrevistados coincidió que el control del trabajo operativo y pericial en las salidas operativas se realiza de forma manual.

En cuanto a las limitantes los directivos consideran que los mayores problemas se manifiestan al no contar con una herramienta informática que les permita manejar la información de forma más eficiente y que realice la gestión de este proceso automáticamente.