

Universidad de las Ciencias Informáticas

Facultad 4



**Sistema para la gestión de la información de la Guardia
Obrero Estudiantil en la Universidad de las Ciencias
Informáticas.**

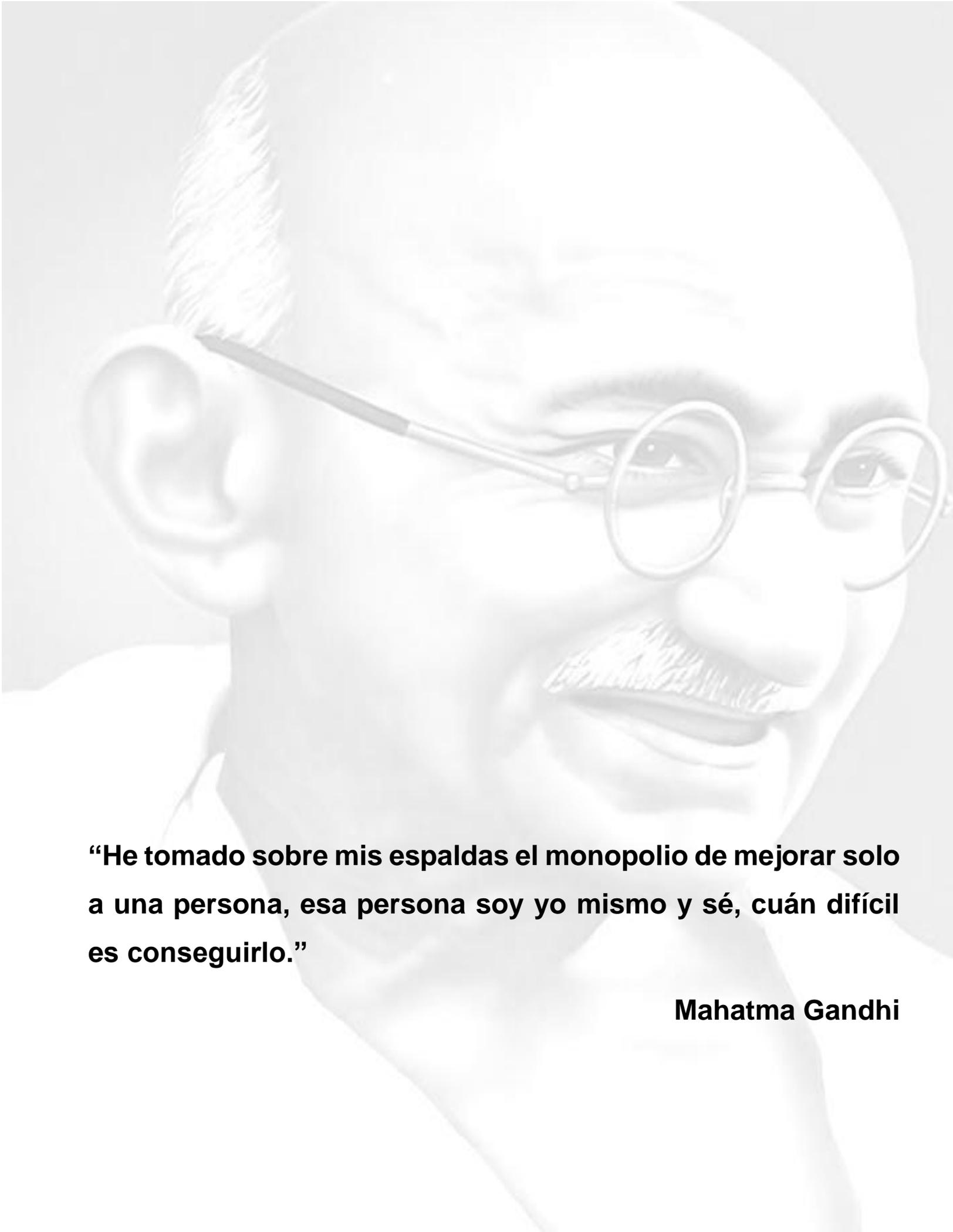
Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Raúl Rosabal Sánchez

Tutor: Ing. Yusimy Rodríguez Ruiz

La Habana, junio 2015.

Año del 57 de la Revolución.



“He tomado sobre mis espaldas el monopolio de mejorar solo a una persona, esa persona soy yo mismo y sé, cuán difícil es conseguirlo.”

Mahatma Gandhi

Declaración de Autoría

Declaro ser el único autor de este trabajo que tiene por título “Sistema para la gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas” y autorizo a la Universidad de las Ciencias Informáticas (UCI) a que haga el uso que estime pertinente con el mismo.

Para que así conste firmo la presente a los _____ días del mes de _____ del año _____.

Autor:

Raúl Rosabal Sánchez

Tutora:

Ing. Yusimy Rodríguez Ruiz

Datos de Contacto

Autor: Raúl Rosabal Sánchez.

Formación Académica: Graduado del Instituto Politécnico de Informática “Rubén Esteban Bravo Álvarez”.

Centro de Estudios: Estudiante de la Universidad de las Ciencias Informáticas (UCI).

Correo Electrónico: rrosabals@estudiantes.uci.cu.

Tutor: Ing. Yusimy Rodríguez Ruiz.

Formación Académica: Graduada en la Universidad de las Ciencias Informáticas.

Centro Laboral: Universidad de las Ciencias Informáticas.

Correo Electrónico: [yrodriguezr@ uci.cu](mailto:yrodriguezr@uci.cu).

Agradecimientos

En primer lugar te agradezco a ti **Dios**, por ayudarme a terminar este proyecto, gracias por darme sabiduría y fortaleza para hacer este sueño realidad.

Quiero agradecer en especial con todo mi amor a la persona que más quiero en la vida, mi **madre**, por ser la persona que siempre ha estado conmigo en todo momento, siempre confiando en mí, ofreciéndome su apoyo incondicional de una forma u otra, guiándome por el camino correcto y haciendo de mí una mejor persona, te amo.

A mis **padres** por los ejemplos de perseverancia y constancia que lo caracterizan y que me ha infundado siempre, por el valor mostrado para salir adelante y por su amor.

A mi **hermano** Reinier por darme su amor y cariño.

A mis **tíos** Laura, Ramiro, Álvaro, Moncy, Tomas y Miriam, por brindarme el calor de una familia cuando yo estaba lejos de la mía todos estos años. Ustedes son para mí como mis segundos padres, los quiero mucho.

A mis **primos** Rafael y Dennis, por estar todo este tiempo presentes ayudándome en todo lo que me hacía falta y quererme tanto como ellos sólo saben hacer.

A mis **abuelitos** Ana y Ramón por estar siempre presente en mis pensamientos y por cuidar de mí.

A todas mis **amistades** de la UCI en especial a: Luis, Alberto, Padilla, Gino, Pedro, Rosbel, Alexis, por su incondicional apoyo durante la realización de este trabajo.

A la **Universidad de las Ciencias Informáticas** por darme la oportunidad de estudiar y ser un profesional.

A mi **tutora** de tesis, Ing. Yusimy Rodríguez Ruiz por su esfuerzo y dedicación, quien con sus conocimientos, su experiencia, su paciencia y su motivación ha logrado en mí que pueda terminar mis estudios con éxito.

También me gustaría agradecer a mis **profesores** durante toda mi carrera profesional porque todos han aportado con un granito de arena a mi formación, y en especial a mis profes Arcadio, Sandra y Ulises por sus consejos, su enseñanza y más que todo por su amistad.

A todos los que de una forma u otra contribuyeron a mi formación como ingeniero informático.

Dedicatoria

Dedico este trabajo de diploma a toda mi familia, especialmente a mis padres y mi hermano, por ser lo máspreciado que tengo en la vida, por su apoyo, consejos, comprensión, amor, ayuda en los momentos difíciles, y por ayudarme con los recursos necesarios para estudiar. Me han dado todo lo que soy como persona, mis valores, mis principios, mi carácter, mi empeño, mi perseverancia, mi coraje para conseguir mis objetivos.

Resumen

En la Universidad de las Ciencias Informáticas la información asociada al proceso de gestión de la información de la Guardia Obrero Estudiantil se realiza de forma manual, convirtiéndolo en un proceso engorroso debido al volumen de información que se maneja. La presente investigación tiene como objetivo desarrollar un sistema informático para mejorar el proceso de gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas. El desarrollo del sistema fue guiado por la metodología de desarrollo Programación Extrema. Como lenguajes de programación del lado del cliente se utilizaron HTML para el maquetado, JavaScript a través de la librería jQuery y CSS apoyado en el framework de diseño Bootstrap. Como lenguaje del lado del servidor se utilizó PHP a través del framework de desarrollo Symfony. Se utilizó el NetBeans IDE como Entorno de Desarrollo Integrado, Apache como servidor web y PostgreSQL como gestor de bases de datos. Se obtuvo como resultado un sistema informático que permite mejorar el proceso de gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas. Además fueron realizadas las pruebas de software a la propuesta de solución, las cuales aseguran en gran medida la correcta implementación de las funcionalidades y un alto grado de satisfacción por parte del cliente.

Palabras claves: gestión de la información, sistema informático.

Introducción	1
Capítulo 1: Fundamentación Teórica	5
1.1 Introducción.....	5
1.2 Principales conceptos asociados con la investigación	5
1.2.1 Información	5
1.2.2 Gestión de la Información	5
1.2.3 Sistemas de Gestión de la Información	6
1.3 Análisis de soluciones existentes	6
1.3.1 Soluciones similares existentes a nivel internacional	6
1.3.2 Soluciones similares existentes a nivel nacional	8
1.3.3 Valoración de los sistemas analizados.....	10
1.4 Metodología de desarrollo	11
1.4.1 Metodología ASD (Adaptive Software Development)	12
1.4.2 Metodología Scrum	13
1.4.3 Metodología XP (Extreme Programming).....	13
1.4.4 Selección de la metodología a utilizar	14
1.5 Lenguaje de modelado	14
1.5.1 UML 2.1	14
1.6 Herramienta de modelado	15
1.6.1 Visual Paradigm 8.0	15
1.7 Lenguajes de programación	15
1.7.1 Lenguajes del lado del cliente	16
1.7.2 Leguajes del lado del servidor.....	17
1.8 Framework	18
1.8.1 Frameworks para la capa de presentación.....	18
1.8.2 Frameworks para la capa de lógica de negocio	20
1.8.3 Frameworks para la capa de acceso a datos	22
1.9 Servidor Web.....	22

1.9.1 Apache 2.4.4.....	23
1.10 Sistema gestor de base de datos	24
1.10.1 PostgreSQL 9.2.4	25
1.11 Entorno de desarrollo integrado.....	26
1.11.1 NetBeans 8.0	26
Conclusiones del capítulo.....	27
Capítulo 2: Construcción de la solución propuesta.....	28
2.1 Introducción.....	28
2.2 Descripción de la solución propuesta	28
2.3 Usuarios relacionados con el sistema.....	29
2.4 Lista de reserva del producto	29
2.5 Historias de usuarios	33
2.6 Planificación	35
2.6.1 Estimación de esfuerzo por historias de usuario	35
2.6.2 Plan de entregas.....	36
2.6.3 Plan de Iteraciones	37
2.7 Diseño del sistema	38
2.7.1 Tarjetas Clase-Responsabilidad-Colaboración (CRC).....	38
2.7.2 Arquitectura	39
2.7.2.1 Patrones de arquitectura.....	39
2.7.2.2 Patrones de diseño	41
2.7.3 Modelo de datos	44
Conclusiones del capítulo.....	44
Capítulo 3: Implementación y pruebas del sistema	46
3.1 Introducción.....	46
3.2 Estándares de codificación.....	46
3.3 Implementación	47
3.3.1 Tareas de Ingeniería.....	47
3.3.1.1 Iteración 1	47
3.3.1.2 Iteración 2	51
3.3.1.3 Iteración 3	52

3.4 Pruebas.....	55
3.4.1 Pruebas unitarias	56
3.4.2 Pruebas de aceptación	57
3.5 Análisis de los resultados de la pruebas.....	62
Conclusiones del capítulo.....	62
Conclusiones generales y Recomendaciones.....	63
Conclusiones generales	63
Recomendaciones.....	63
Referencias Bibliográficas.....	64
Anexos.....	68
Anexo 1: Historias de Usuario	68
Anexo 2: Tarjetas CRC.....	71
Anexo 3: Tareas de Ingeniería	72
Anexo 4: Pruebas de Aceptación	80

Índice de Tablas

Tabla 1: Ranking de “agilidad” de las metodologías (17).....	12
Tabla 2: Usuarios del sistema.	29
Tabla 3: LRP-Requisitos Funcionales.	29
Tabla 4: LRP-Requisitos no Funcionales.	31
Tabla 5: HU-Autenticar usuario.	33
Tabla 6: HU-Gestionar usuario.....	33
Tabla 7: HU-Gestionar área.	34
Tabla 8: HU-Gestionar hechos delictivos.	34
Tabla 9: HU-Generar informe.....	34
Tabla 10: Estimación de esfuerzo por HU.	35
Tabla 11: Plan de entregas.	36
Tabla 12: Funcionalidades disponibles por entrega del producto.	36
Tabla 13: Plan de iteraciones.....	37
Tabla 14: Tarjeta CRC-Usuario.....	38
Tabla 15: Tarjeta CRC-Área.	38
Tabla 16: Tarjeta CRC-Hechos delictivos.	39
Tabla 17: Tarjeta CRC-Parte diario.....	39
Tabla 18: Historias de Usuario implementadas en la 1ra Iteración.....	47
Tabla 19: Tarea 1 de la HU Autenticar usuario.	48
Tabla 20: Tarea 2 de la HU Gestionar usuario.	48
Tabla 21: Tarea 3 de la HU Gestionar usuario.	48
Tabla 22: Tarea 4 de la HU Gestionar usuario.	49
Tabla 23: Tarea 5 de la HU Gestionar usuario.	49
Tabla 24: Tarea 6 de la HU Gestionar usuario.	49
Tabla 25: Tarea 8 de la HU Gestionar área.....	49
Tabla 26: Tarea 9 de la HU Gestionar área.....	50
Tabla 27: Tarea 10 de la HU Gestionar área.....	50
Tabla 28: Tarea 11 de la HU Gestionar área.....	50
Tabla 29: Tarea 12 de la HU Gestionar área.....	50
Tabla 30: Historias de Usuario implementadas en la 2da Iteración.	51

Tabla 31: Tarea 23 de la HU Gestionar hechos delictivos.....	51
Tabla 32: Tarea 24 de la HU Gestionar hechos delictivos.....	51
Tabla 33: Tarea 25 de la HU Gestionar hechos delictivos.....	52
Tabla 34: Tarea 26 de la HU Gestionar hechos delictivos.....	52
Tabla 35: Tarea 27 de la HU Gestionar hechos delictivos.....	52
Tabla 36: Historias de Usuario implementadas en la 3ra Iteración.....	52
Tabla 37: Tarea 48 de la HU Gestionar guardia.....	53
Tabla 38: Tarea 49 de la HU Gestionar guardia.....	53
Tabla 39: Tarea 50 de la HU Gestionar guardia.....	53
Tabla 40: Tarea 51 de la HU Gestionar guardia.....	54
Tabla 41: Tarea 52 de la HU Gestionar guardia.....	54
Tabla 42: Tarea 53 de la HU Generar informe.....	54
Tabla 43: Tarea 54 de la HU Generar reportes.....	54
Tabla 44: Tarea 55 de la HU Consultar estadísticas.....	55
Tabla 45: Tarea 56 de la HU Exportar a PDF.....	55
Tabla 46: Métodos del framework PHPUnit para las pruebas unitarias (52).....	56
Tabla 47: Caso de prueba de aceptación #1.....	57
Tabla 48: Caso de prueba de aceptación #2.....	57
Tabla 49: Caso de prueba de aceptación #3.....	58
Tabla 50: Caso de prueba de aceptación #4.....	58
Tabla 51: Caso de prueba de aceptación #5.....	58
Tabla 52: Caso de prueba de aceptación #6.....	59
Tabla 53: Caso de prueba de aceptación #7.....	59
Tabla 54: Caso de prueba de aceptación #8.....	59
Tabla 55: Caso de prueba de aceptación #9.....	60
Tabla 56: Caso de prueba de aceptación #18.....	60
Tabla 57: Caso de prueba de aceptación #19.....	60
Tabla 58: Caso de prueba de aceptación #20.....	61
Tabla 59: Caso de prueba de aceptación #21.....	61
Tabla 60: HU-Asignar rol.....	68
Tabla 61: HU-Gestionar acercamiento de extranjeros.....	68

Tabla 62: HU-Gestionar incidencias.....	68
Tabla 63: HU-Gestionar violaciones.....	68
Tabla 64: HU-Gestionar afectaciones.	69
Tabla 65: HU-Gestionar visitas.	69
Tabla 66: HU-Gestionar extranjero.	69
Tabla 67: HU-Gestionar guardia.	70
Tabla 68: HU-Generar reportes.....	70
Tabla 69: HU-Consultar estadísticas.....	70
Tabla 70: HU-Exportar a PDF.	71
Tabla 71: Tarjeta CRC-Acercamiento de extranjeros.	71
Tabla 72: Tarjeta CRC-Incidencias.	71
Tabla 73: Tarjeta CRC-Violaciones.....	71
Tabla 74: Tarjeta CRC-Afectaciones.....	71
Tabla 75: Tarjeta CRC-Visitas.....	71
Tabla 76: Tarjeta CRC-Extranjero.....	72
Tabla 77: Tarjeta CRC- Guardia.	72
Tabla 78: Tarea 7 de la HU Asignar rol.	72
Tabla 79: Tarea 13 de la HU Gestionar Acercamiento de extranjeros.....	72
Tabla 80: Tarea 14 de la HU Gestionar Acercamiento de extranjeros.....	72
Tabla 81: Tarea 15 de la HU Gestionar Acercamiento de extranjeros.....	73
Tabla 82: Tarea 16 de la HU Gestionar Acercamiento de extranjeros.....	73
Tabla 83: Tarea 17 de la HU Gestionar Acercamiento de extranjeros.....	73
Tabla 84: Tarea 18 de la HU Gestionar incidencias.	74
Tabla 85: Tarea 19 de la HU Gestionar incidencias.	74
Tabla 86: Tarea 20 de la HU Gestionar incidencias.	74
Tabla 87: Tarea 21 de la HU Gestionar incidencias.	74
Tabla 88: Tarea 22 de la HU Gestionar incidencias.	75
Tabla 89: Tarea 28 de la HU Gestionar violaciones.	75
Tabla 90: Tarea 29 de la HU Gestionar violaciones.	75
Tabla 91: Tarea 30 de la HU Gestionar violaciones.	75
Tabla 92: Tarea 31 de la HU Gestionar violaciones.	76

Tabla 93: Tarea 32 de la HU Gestionar violaciones.	76
Tabla 94: Tarea 33 de la HU Gestionar afectaciones.	76
Tabla 95: Tarea 34 de la HU Gestionar afectaciones.	76
Tabla 96: Tarea 35 de la HU Gestionar afectaciones.	77
Tabla 97: Tarea 36 de la HU Gestionar afectaciones.	77
Tabla 98: Tarea 37 de la HU Gestionar afectaciones.	77
Tabla 99: Tarea 38 de la HU Gestionar visitas.	78
Tabla 100: Tarea 39 de la HU Gestionar visitas.	78
Tabla 101: Tarea 40 de la HU Gestionar visitas.	78
Tabla 102: Tarea 41 de la HU Gestionar visitas.	78
Tabla 103: Tarea 42 de la HU Gestionar visitas.	78
Tabla 104: Tarea 43 de la HU Gestionar extranjero.	79
Tabla 105: Tarea 44 de la HU Gestionar extranjero.	79
Tabla 106: Tarea 45 de la HU Gestionar extranjero.	79
Tabla 107: Tarea 46 de la HU Gestionar extranjero.	79
Tabla 108: Tarea 47 de la HU Gestionar extranjero.	80
Tabla 109: Caso de prueba de aceptación #10.	80
Tabla 110: Caso de prueba de aceptación #11.	80
Tabla 111: Caso de prueba de aceptación #12.	81
Tabla 112: Caso de prueba de aceptación #13.	81
Tabla 113: Caso de prueba de aceptación #14.	82
Tabla 114: Caso de prueba de aceptación #15.	82
Tabla 115: Caso de prueba de aceptación #16.	82
Tabla 116: Caso de prueba de aceptación #17.	83
Tabla 117: Caso de prueba de aceptación #22.	83
Tabla 118: Caso de prueba de aceptación #23.	83
Tabla 119: Caso de prueba de aceptación #24.	84
Tabla 120: Caso de prueba de aceptación #25.	84
Tabla 121: Caso de prueba de aceptación #26.	84
Tabla 122: Caso de prueba de aceptación #27.	85
Tabla 123: Caso de prueba de aceptación #28.	85

Tabla 124: Caso de prueba de aceptación #29.....	86
Tabla 125: Caso de prueba de aceptación #30.....	86
Tabla 126: Caso de prueba de aceptación #31.....	86
Tabla 127: Caso de prueba de aceptación #32.....	87
Tabla 128: Caso de prueba de aceptación #33.....	87
Tabla 129: Caso de prueba de aceptación #34.....	87
Tabla 130: Caso de prueba de aceptación #35.....	88
Tabla 131: Caso de prueba de aceptación #36.....	88
Tabla 132: Caso de prueba de aceptación #37.....	88
Tabla 133: Caso de prueba de aceptación #38.....	89
Tabla 134: Caso de prueba de aceptación #39.....	89
Tabla 135: Caso de prueba de aceptación #40.....	90
Tabla 136: Caso de prueba de aceptación #41.....	90

Índice de Figuras

Figura 1: Fases del ciclo de vida de la metodología XP.	14
Figura 2: Estadísticas para sitios web usando la tecnología jQuery (35).	19
Figura 3: Cuota de mercado de las librerías JavaScript más populares en Internet (36).	19
Figura 4 Cuota de mercado de los frameworks web más populares en Internet (38).	20
Figura 5: Estadísticas para sitios web usando servidor web (43).	23
Figura 6: Cuota de mercado de los servidores web más populares en Internet (44).	24
Figura 7: Funcionamiento básico del patrón MVC en Symfony.	40
Figura 8: Declaración de la clase Área.....	42
Figura 9: Función donde se crea un objeto de la clase Área.....	42
Figura 10: Patrón Decorator en Symfony.	44
Figura 11: Modelo de datos.	44
Figura 12: Total de funcionalidades probadas y no conformidades encontradas por iteración.	62

Introducción

Resulta innegable el auge cada vez mayor de las Tecnologías de la Información y las Comunicaciones (TIC's) en las diferentes esferas de la sociedad a escala mundial. El desarrollo impetuoso de la ciencia y la tecnología ha llevado a la sociedad a entrar al nuevo milenio inmerso en lo que se ha dado en llamar "era de la información" (1), generando nuevos cambios que condicionan y determinan la vida ciudadana.

Para las empresas e instituciones que manejan gran cantidad de recursos humanos y materiales, garantizar la seguridad de los mismos se convierte en una ardua tarea, en la cual se invierten cuantiosos recursos en el uso de nuevas tecnologías con estos fines. Con las TIC's surgieron nuevas soluciones con el fin de satisfacer las necesidades elementales de las instituciones, de preservar sus recursos humanos y materiales; estableciendo el uso de sistemas para la seguridad y protección (2).

La Universidad de las Ciencias Informática (UCI) es una de las instituciones que necesita garantizar la seguridad y protección de los recursos humanos y materiales que posee. Desde la creación de la misma se hizo indispensable llevar a cabo una estrategia para resguardar todos los recursos con que cuenta. Se implementó un sistema de Guardia Obrero Estudiantil (GOE) en donde participan, como su nombre indica, todos los trabajadores y estudiantes del Centro divididos en diferentes áreas, además de un puesto de mando a nivel UCI.

La información asociada al proceso de gestión de la información de la Guardia Obrero Estudiantil en la UCI actualmente se realiza de forma manual, convirtiéndolo en un proceso engorroso, debido al volumen de información que se maneja, el número de personas que manipulan los datos y la cantidad de cálculos que se realizan. Esta situación posibilita que ocurran demoras, pérdidas y/o duplicidad de la información, lo que le dificulta, a los especialistas, la obtención de reportes y estadísticas precisas sobre el comportamiento de la Guardia Obrera Estudiantil en la Universidad. Por otra parte la UCI posee los medios tecnológicos y el personal capacitado para mejorar este proceso, sin embargo no emplean las potencialidades de las TIC's puestas a su disposición de manera efectiva para lograr este objetivo.

Partiendo de la situación problemática antes descrita, se plantea como **problema a resolver**: ¿Cómo mejorar el proceso de gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas?

Se define como **objeto de estudio** el proceso de gestión de la información, y el **campo de acción** en que se enmarca el trabajo es el proceso de gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas.

Por tales razones el **objetivo general** es desarrollar un Sistema informático para mejorar el proceso de gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas.

Para darle cumplimiento al objetivo general se proponen los siguientes **objetivos específicos**:

- Construir los referentes teóricos fundamentales que sustentan la investigación.
- Diseñar el Sistema para la gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias informáticas.
- Implementar el Sistema para la gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias informáticas.
- Validar el Sistema para la gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias informáticas mediante pruebas de software.

Se plantea la **hipótesis** de que si se desarrolla un sistema informático se mejorará el proceso de gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas.

El posible **resultado** a lograr es un sistema informático para la gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas.

Para garantizar el cumplimiento de los objetivos específicos se definen las siguientes **tareas de investigación**:

- Análisis de los fundamentos teóricos y metodológicos que sustentan la creación del Sistema para la gestión de información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas.
- Búsqueda de información sobre soluciones similares a nivel nacional e internacional.
- Análisis y selección de herramientas, tecnologías y metodologías para el desarrollo de la solución.
- Elaboración de los artefactos en correspondencia con la metodología seleccionada.
- Diseño de la base de datos que soporte el sistema.
- Implementación de la solución propuesta.
- Validación de la solución mediante pruebas de software.

Tomando como punto de partida el método dialéctico –materialista, durante el desarrollo de la investigación se usaron métodos de orden teórico y de orden empírico:

Métodos teóricos:

- **Analítico –Sintético:** Se utilizó para analizar y comprender la teoría y documentación de distintos autores relacionada con el tema de investigación, permitiendo así, extraer los elementos más coherentes e importantes relacionados con el proceso de gestión de la información.
- **Histórico –Lógico:** Se empleó para el estudio de la evolución de los procesos vinculados con la gestión de información y las tendencias actuales para el desarrollo del Sistema para la gestión de información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas.

Métodos empíricos:

- **Entrevista:** Se utilizó para conocer las necesidades del cliente y obtener los requisitos del sistema a tener en cuenta para la solución de la investigación.
- **Observación:** Se empleó para el estudio de las herramientas similares con el objetivo de obtener una idea de cómo realizar el diseño de la interfaz visual y comprender el funcionamiento de los sistemas de gestión de la información.

La estructura de la presente investigación está compuesta por tres capítulos, los cuales se describen a continuación:

Capítulo 1: Fundamentación Teórica

Se presentan los elementos teóricos que sustentan la investigación. Se realiza un estudio del estado del arte de diferentes sistemas existentes para la guardia a nivel nacional e internacional. Además se realiza un análisis detallado de las herramientas, metodologías y tecnologías a utilizar en el proceso de desarrollo del sistema y la fundamentación de la selección de las más adecuadas para ser usada en el desarrollo de la propuesta de solución al problema actual.

Capítulo 2: Construcción de la solución propuesta

Se describe la propuesta de solución del problema planteado, detallando las fases y artefactos de acuerdo con la metodología seleccionada. Además se realiza el diseño del sistema.

Capítulo 3: Implementación y pruebas del sistema

Se presentan los aspectos relacionados con la implementación de la solución propuesta. Además se realiza la validación de los resultados obtenidos mediante pruebas unitarias y pruebas de aceptación.

Al final se podrán observar las conclusiones generales, además de las recomendaciones y las referencias bibliográficas hechas a lo largo del trabajo. Por último, se podrán observar los anexos, los cuales contienen información de apoyo a algunos aspectos tratados durante la investigación.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se describen los elementos teóricos que sustentan la investigación. Se realiza un estudio sobre las soluciones similares existentes que puedan solucionar el problema planteado, así como las tendencias, herramientas, tecnologías y metodologías que se utilizaron en el desarrollo de la investigación.

1.2 Principales conceptos asociados con la investigación

1.2.1 Información

Es un conjunto de datos acerca de algún suceso, hecho, fenómeno o situación, que organizados en un contexto determinado tienen un significado, cuyo propósito puede ser el de reducir la incertidumbre o incrementar el conocimiento acerca de algo (3). Por otro lado el investigador Geoffrey Hirt agrega además que los mismos se encuentran disponibles para su uso inmediato y están estrictamente ligados a la toma de decisiones (4).

Agrupando estos conceptos, se puede decir que la información no es más que un conjunto de datos organizados en un contexto determinado, que tiene un significado y permite a quien la usa, incrementar su conocimiento y reducir incertidumbres, lo que contribuye a la toma de decisiones.

1.2.2 Gestión de la Información

Según la investigadora MsC. María Josefina Vidal Ledo, la gestión de la información es el proceso de organizar, evaluar, presentar, comparar los datos en un determinado contexto, controlando su calidad, de manera que esta sea veraz, oportuna, significativa, exacta y útil, y que esté disponible en el momento que se necesite. Se encamina al manejo de la información, documentos, metodologías, informes, publicaciones, soportes y flujos en función de los objetivos estratégicos de una organización (5). Mediante la gestión de la información se proporcionan los recursos de información necesarios para una buena toma de decisiones, se desarrollan nuevos conocimientos que posibilitan calidad y eficiencia en los servicios y productos de las organizaciones (6).

Se concluye que la gestión de la información es el proceso de organizar, evaluar, comparar, eliminar y divulgar la información existente en un contexto determinado, garantizando su integridad, confidencialidad y disponibilidad en el momento que sea necesario, propiciando los recursos necesarios para una buena toma de decisiones.

1.2.3 Sistemas de Gestión de la Información

Es un conjunto de sistemas y procedimientos que recopilan información de una variedad de fuentes, la compilan y la presentan en un formato legible. Los sistemas actuales de gestión de la información se basan en gran medida en la tecnología para recopilar y presentar datos, pero el concepto es más antiguo que las tecnologías informáticas modernas. El propósito principal de un sistema de gestión de la información es hacer que la toma de decisiones por parte de los especialistas sea más eficiente y productiva (7). Los sistemas de gestión de información constituyen hoy una alternativa de imprescindible presencia en cada organización al permitir operar casi todos los activos tangibles e intangibles de la institución y llegar a convertirse en la herramienta integral de gerencia más cotizada y necesaria para alcanzar con éxito los resultados propuestos por la organización (8).

Por lo tanto un sistema de gestión de la información puede considerarse como una herramienta que permite la gestión de los recursos de información de una organización. Su finalidad es generar servicios que respondan a las necesidades de los usuarios, aprovechando al máximo sus recursos en función de la mejora continua y de la toma de decisiones.

1.3 Análisis de soluciones existentes

La evolución en el mundo del software ha llevado a una gran competencia en cuanto a los sistemas informatizados. Los sistemas para la gestión de la Guardia no están libres de esta competitividad y se han desarrollado de maneras muy diversas, lo que ha permitido la creación de productos de muy buena calidad y un alto grado de compromiso. A continuación se analizan algunas soluciones similares existentes tanto en el ámbito nacional como en el ámbito internacional.

1.3.1 Soluciones similares existentes a nivel internacional

Kid de Control de Rondas: Es un sistema que permite lograr con efectividad el control de recorridos de cualquier persona de seguridad y de todo trabajo rutinario o mantenimiento, logrando obtener luego,

informes que permitirán evaluar si el recorrido se ha cumplido, en qué fecha y a qué hora estuvo esa persona presente en cada punto de control que se haya establecido (9).

Entre sus principales características se encuentran:

- Ideal para la vigilancia y control de cumplimientos de rutinas.
- Su funcionamiento es totalmente electrónico y sin partes móviles.
- Identificación de puestos por nombre y número de ID.
- Informes agrupados por fecha, colectores, puestos, grupos y objetivos.
- El sistema permite exportar los informes en diferentes formatos, incluidos Word¹, Excel², entre otros.
- El sistema genera un correo electrónico para distribuir los informes del control de la ronda a los distintos usuarios.

Sistema control de rondas para guardias vigilantes rondines: Este sistema tiene como función principal registrar las rondas que un guardia debe efectuar durante la vigilancia de un recinto, almacenando en un dispositivo la dirección o localización del punto visitado, la fecha y la hora en que ocurrió dicho evento. Este sistema permite (10):

- Registrar las marcas efectuadas por un guardia durante su ronda.
- Analizar las marcaciones a fin de determinar si han existido desviaciones sobre la norma establecida en cuanto a los puntos visitados, la secuencia en que fueron visitados los puntos y los tiempos empleados entre la visita a cada punto definido.
- Emitir informes sobre las desviaciones y los eventos registrados.

Control de rondas de vigilancia mediante NFC³: Este sistema permite tener localizados a los vigilantes de seguridad y saber en qué punto de la ronda se encuentran. Cada vigilante debe seguir una programación regular de las rondas y respetar algunos procedimientos para cumplir con el compromiso hacia el cliente, además de aumentar su propia seguridad en la realización de sus actividades (11).

Ventajas que proporciona el Sistema de control de rondas:

- Rondas de vigilancia notablemente más eficaces.

¹ Aplicación distribuida por Microsoft Office para procesar textos.

² Aplicación distribuida por Microsoft Office para hojas de cálculo.

³ Del inglés Near Field Communication, en español comunicación de campo cercano.

- Una planificación más fácil.
- Un tiempo de respuesta más rápido en caso de emergencia.
- Comprobación de la finalización de la ronda tanto para la empresa como para el cliente final.
- Fácil generación de informes para nóminas y contabilidad.
- Formación eficiente para asignar guardias a determinadas rondas.

1.3.2 Soluciones similares existentes a nivel nacional

Gestor Web para el control de la Guardia Obrera de la Universidad de las Ciencias Informáticas: Este sistema se desarrolla con el objetivo de mejorar el proceso de control de la Guardia Obrera de la Universidad de las Ciencias Informáticas, logrando obtener luego, informes que permitirán evaluar el comportamiento de la Guardia Obrera (12).

Entre sus principales características se encuentran:

- Es independiente del sistema operativo donde se ejecute y presenta un requerimiento de hardware mínimo.
- Brinda una amplia información sobre todo el tema perteneciente al control de la Guardia Obrera.
- Disminuye el trabajo manual y agiliza el flujo de información, ahorrando así tiempo y recursos a la universidad.
- Ofrece la posibilidad de realizar reportes semanalmente relacionados con todo el proceso de control de la guardia.
- Cuenta con una interfaz web amigable y fácil de usar permitiendo que los usuarios puedan disfrutar de sus servicios.

En la actualidad este sistema no está en funcionamiento, aunque incluye dentro de sus funcionalidades el control de la Guardia Obrera, se centra solamente en este proceso. Este sistema no se rige por las nuevas normas establecidas para la realización de la GOE, ya que esta planificación es solo para los trabajadores sin tener en cuenta a los estudiantes. Por lo tanto, no toma en cuenta todas las funcionalidades necesarias para la gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas.

Sistema de Guardia Estudiantil de la Facultad 3: Este sistema se desarrolla con el objetivo de mejorar el proceso de planificación de la Guardia Estudiantil en la facultad 3 de la Universidad de las Ciencias Informáticas.

Entre sus principales características se encuentran (13):

- Permite generar la planificación de la guardia.
- Permite configurar las postas y los turnos.
- Realiza la planificación teniendo en cuenta criterios como: no repetir los turnos, planificar primero a los evaluados de Mal del mes anterior, planificar seguidamente a los que no hayan realizado la guardia y después al resto.
- Notifica la planificación por correo electrónico y por el Jabber⁴.
- Permite evaluar, gestionar las guardias y los usuarios.
- Les facilita a los estudiantes visualizar cuándo les toca su guardia y sus evaluaciones.

Este sistema se centra solamente en el proceso de planificación de la Guardia Estudiantil. Dentro de sus funcionalidades incluye la generación de la guardia mediante la utilización de un método bayesiano, con un tiempo de respuesta del sistema de 5 minutos al realizar la planificación de un mes. Este sistema no se rige por las nuevas normas establecidas para la realización de la GOE, ya que esta planificación es solo para los estudiantes sin tener en cuenta a los trabajadores. Por lo tanto, no toma en cuenta todas las funcionalidades necesarias para la gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas.

Sistema de Información Geográfica para la Dirección de Seguridad y Protección en la Universidad de las Ciencias Informáticas: Este sistema desarrollado en el 2013, permite representar espacialmente la información de los procesos de planificación y control de la Guardia Obrera Estudiantil en la Dirección de Seguridad y Protección de la Universidad de las Ciencias Informáticas (2).

Entre sus principales características se encuentran:

- Permite representar visualmente la planificación de la guardia.
- Permite visualizar las postas cuyo personal de guardia está incompleto.

⁴ Protocolo abierto y libre para mensajería instantánea.

- Permite configurar las áreas, las postas y los turnos.
- Permite publicar la guardia planificada.
- Exportar la planificación de guardia.
- Importar datos de un recurso humano en el sistema.
- Identificar activos sobre el mapa.

Este sistema se centra en la representación espacial del proceso de planificación de la Guardia Obrera Estudiantil en la Dirección de Seguridad y Protección de la Universidad de las Ciencias Informáticas, no toma en cuenta todas las funcionalidades necesarias para la gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas.

1.3.3 Valoración de los sistemas analizados

Las investigaciones realizadas hasta el momento arrojaron como resultado que no existe ningún sistema que cumpla con todas las características definidas para la gestión de la información de la GOE en la UCI. Las soluciones analizadas a nivel internacional son todas de carácter propietario, lo que implica que hay que pagar para adquirir su licencia. Sin embargo, el estudio de estos sistemas proporcionó una serie de características que pueden ser agregadas a la investigación, entre las que se encuentran:

- Permiten lograr con gran efectividad el control de recorridos de cualquier persona de seguridad.
- Permiten generar informes que permiten evaluar el recorrido.
- Permiten registrar las rondas que un Guardia debe efectuar durante la vigilancia.

El análisis de los sistemas a nivel nacional arrojó que existía un sistema con el objetivo de informatizar el proceso de control de la Guardia Obrera Estudiantil en la Universidad de las Ciencias Informáticas. El mismo, luego de realizar una encuesta no se utilizaba, ya que era una aplicación de escritorio de gran tamaño, lo que dificultó su instalación y ejecución en las computadoras con que cuentan las áreas. Además posee un gran número de funcionalidades, pero en su mayoría no son útiles a los especialistas de control de la GOE y carece de otras que sí necesitan.

Por tal motivo se decide desarrollar un sistema informático que mejore el proceso de gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas y que a la vez pueda funcionar desde cualquier computadora de la Universidad.

1.4 Metodología de desarrollo

Una metodología de desarrollo en la ingeniería de software es un conjunto de procedimientos, técnicas, herramientas y soporte documental que debe seguirse para el desarrollo del software, proporcionando una ayuda muy importante e indispensable para que el producto final posea las funcionalidades requeridas por el cliente y que cumpla con las necesidades del mismo y del usuario final (14).

El análisis de la bibliografía consultada, muestra que existen dos tipos de metodologías de desarrollo de software, las metodologías robustas o tradicionales y las metodologías ágiles.

Metodologías tradicionales

Las metodologías tradicionales son conocidas también como metodologías pesadas. Centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto definido en la fase inicial del desarrollo del proyecto. Otra de las características importantes dentro de este enfoque, son los altos costes al implementar un cambio y la falta de flexibilidad en proyectos donde el entorno es volátil. Estas metodologías han demostrado ser bastante efectivas en proyectos de gran tamaño, donde el equipo de desarrollo es grande y con una amplia experiencia en su aplicación (15).

Metodologías ágiles

Las metodologías ágiles han surgido para aportar valor a los usuarios en contraposición a las metodologías tradicionales que buscan ser fieles a las planificaciones iniciales de los proyectos, adaptándose a los cambios y evolucionando de forma conjunta con el software. Las metodologías ágiles son flexibles, sus proyectos son subdivididos en proyectos más pequeños, incluyen comunicación constante con el cliente, son altamente colaborativos y se adaptan mejor a los cambios. De hecho, el cambio en los requerimientos es una característica esperada al igual que las entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente. Además se caracterizan por el desarrollo iterativo e incremental; la simplicidad de la implementación; las entregas frecuentes; la priorización de los requerimientos o características a desarrollar a cargo del cliente; y la cooperación entre desarrolladores y clientes (16).

La propuesta de solución a desarrollar posee características que hacen que se ajuste a una metodología ágil debido a que es corto periodo de tiempo para el desarrollo, la dimensión de la solución es pequeña, los requisitos pueden sufrir cambios constantemente y el cliente forma parte del equipo de desarrollo. A

continuación se selecciona, dentro de las metodologías ágiles, cuál es la más adecuada para ser utilizada en el presente trabajo. Para ello se muestra un estudio de las principales metodologías de esta clasificación.

La Tabla 1 compara distintas aproximaciones ágiles en base a tres parámetros: vista del sistema como algo cambiante, tener en cuenta la colaboración entre los miembros del equipo y características más específicas de la propia metodología como son simplicidad, excelencia técnica, resultados, adaptabilidad.

Tabla 1: Ranking de “agilidad” de las metodologías (17).

	CMM ⁵	ASD ⁶	Crystal	DSDM ⁷	FDD ⁸	LD ⁹	Scrum	XP ¹⁰
Sistema como algo cambiante	1	5	4	3	3	4	5	5
Colaboración	2	5	5	4	4	4	5	5
Resultados	2	5	5	4	4	4	5	5
Simplicidad	1	4	4	3	5	3	5	5
Adaptabilidad	2	5	5	3	3	4	4	3
Excelencia técnica	4	3	3	4	4	4	3	4
Prácticas de colaboración	2	5	5	4	3	3	4	5
Media Total	1.7	4.8	4.5	3.6	3.6	3.9	4.7	4.8

1.4.1 Metodología ASD (Adaptive Software Development)

El desarrollo adaptativo de software (ASD) lo propuso Jim Highsmith 1998 como una técnica para construir software y sistemas complejos. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo (18).

⁵ Del inglés Capability Maturity Model, en español Modelo de Capacidad y Madurez.

⁶ Del inglés Adaptive Software Development, en español Desarrollo de Software Adaptable.

⁷ Del inglés Dynamic Systems Development Method, en español Método de Desarrollo de Sistemas Dinámicos.

⁸ Del inglés Feature Driven Development, en español Desarrollo Basado en Funcionalidades.

⁹ Del inglés Lean Development, en español Desarrollo Ajustado.

¹⁰ Del inglés Extreme Programming, en español Programación Extrema.

1.4.2 Metodología Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras, y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos. Está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados con rapidez, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad y la productividad son fundamentales. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente, cuando este lo solicite (19).

1.4.3 Metodología XP (Extreme Programming)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes (20).

Entre las características principales de esta metodología se encuentran:

- Metodología creada a base de prueba y error.
- Énfasis en el desarrollo del software más que una buena documentación.
- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- No introduce funcionalidades antes de que sean necesarias.
- El cliente se integra como miembro del equipo de desarrollo.

El ciclo de vida ideal de XP consta de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Cierre de Proyecto (21). Este ciclo puede apreciarse en la Figura 1.

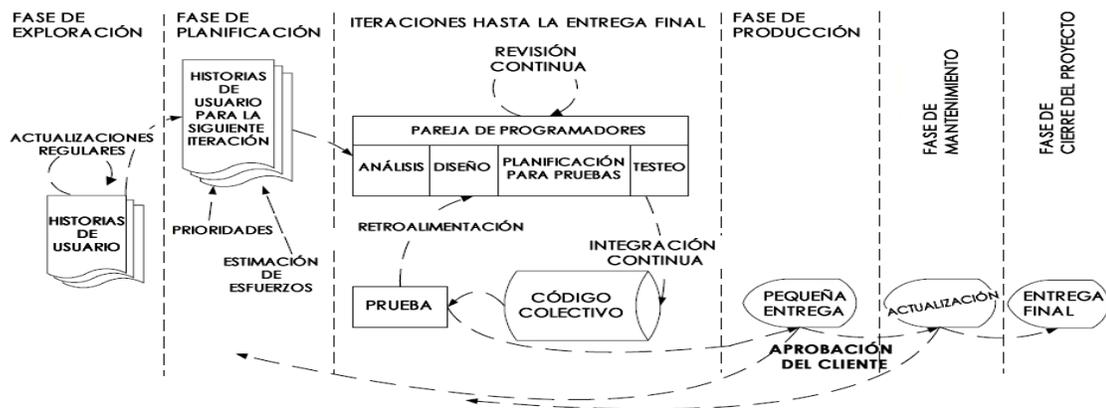


Figura 1: Fases del ciclo de vida de la metodología XP.

1.4.4 Selección de la metodología a utilizar

Se selecciona XP para guiar el proceso de desarrollo de software de la propuesta solución porque las características que describen a esta metodología son las que más se adaptan al entorno de desarrollo del sistema. Algunas de estas características son: el cliente forma parte del equipo de desarrollo, las entregas de las iteraciones son de una a tres semanas, la dimensión de la solución es pequeña y se cuenta con un corto período de tiempo para desarrollar la aplicación.

1.5 Lenguaje de modelado

1.5.1 UML 2.1

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language), es un lenguaje gráfico que se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. Es un lenguaje de propósito general para el modelado orientado a objetos. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, pues ha sido diseñado para modelar los más variados tipos de negocio (22).

Luego de haber seleccionado el lenguaje de modelado, se procederá al análisis de las herramientas que soportan UML para seleccionar la más adecuada.

1.6 Herramienta de modelado

Una herramienta de modelado es un conjunto de 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 ayudan 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 costes, implementación automática de parte del código con el diseño dado, compilación automática y documentación o detección de errores (23).

1.6.1 Visual Paradigm 8.0

Visual Paradigm es una herramienta profesional para UML que soporta el ciclo de vida completo del desarrollo de software. Permite generar código inverso, generar código desde diagramas y documentación. Es muy popular por sus disímiles características, producto de calidad que soporta aplicaciones web, muy fácil de instalar y actualizar. Permite la generación de código para varios lenguajes. 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 (24). Entre sus características se destacan las siguientes:

- Soporta aplicaciones web.
- Fácil de instalar y actualizar.
- Fácil aprendizaje.
- Abundante documentación.
- Multiplataforma, robusta, íntegra y gratuita.

Luego de seleccionada la herramienta, es necesario escoger el lenguaje de programación a utilizar para la implementación del sistema.

1.7 Lenguajes de programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos (25). Los lenguajes de programación para la Web se dividen en dos grupos: los lenguajes del lado del cliente y los lenguajes del lado del servidor.

1.7.1 Lenguajes del lado del cliente

Los lenguajes del lado del cliente son totalmente independientes del servidor, se ejecutan en el navegador del usuario y se ven como páginas dinámicas que se procesan en el cliente.

HTML 5

La W3C¹¹ define el Lenguaje de Marcado de Hipertexto (HTML, de sus siglas en inglés HyperText Markup Language) como un lenguaje comúnmente utilizado para la publicación de hipertexto en la Web y desarrollado con la idea de que cualquier persona o tipo de dispositivo pueda acceder a la información en la Web. HTML utiliza etiquetas que marcan elementos y estructuran el texto de un documento (26).

La última versión de este lenguaje es la conocida HTML 5, la misma está pensada con una mayor integración con los lenguajes CSS y JavaScript. HTML 5 ha revolucionado la web y no solo se ve como el presente, sino como el futuro, por las numerosas novedades que trae con respecto a la versión anterior, entre las que se encuentran (27):

- Incluye nuevas etiquetas que permiten representar elementos familiares de las páginas, tal es el caso de: `<header>` para el encabezado de las páginas, `<nav>` para la navegación, `<section>` para una sección de la página y `<figure>` para asignar un título a una imagen.
- Incluye etiquetas para incorporar contenido multimedia como `<audio>` para audio y `<video>` para video, las cuales permiten la reproducción por parte del navegador de este tipo de contenido.
- Incorpora nuevas APIs¹² que facilitan en gran medida el trabajo de los desarrolladores, entre las que destacan la Geolocalización, *Drag and Drop* y el trabajo con Bases de Datos locales.

CSS 3

Las hojas de estilo en cascada, CSS por sus siglas en inglés, (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos (28).

¹¹ Del inglés World Wide Web Consortium.

¹² Del inglés Application Programming Interface.

La última versión del lenguaje CSS es la versión 3 (CSS3), la cual ha incorporado valiosas novedades, como (29):

- Nuevas alternativas para dibujar bordes con el uso de opciones tales como color, imágenes, y radio o redondeado.
- Novedades en el trabajo con fondos, con el uso de degradados y la posibilidad de incluir múltiples imágenes.
- Uso de sombras para texto.
- Se incluye la posibilidad de aplicar sombra a los elementos.
- Novedades en cuanto al uso del color y de la opacidad.
- Incorporación de muchas novedades en lo que se refiere a flujo de texto dentro del sitio.
- Nuevas características para el trabajo con múltiples columnas.
- Características relacionadas con la interfaz de usuario.
- Capacidad de rotación de elementos.
- Opciones de transformación de elementos.
- Incorporación de transición y también funciones de animación.

JavaScript 1.5

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como aparición y desaparición de texto, animaciones, acciones que se activan al pulsar botones u otros elementos y ventanas con mensajes de aviso al usuario. Técnicamente JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (30).

1.7.2 Leguajes del lado del servidor

Los lenguajes del lado del servidor se ejecutan del lado del servidor, consisten en el procesamiento de una petición de un usuario mediante la interpretación de un script¹³ en el servidor web para generar páginas HTML como respuesta.

¹³ Es un programa usualmente simple, que por lo general se almacena en un archivo de texto plano.

PHP 5.4

PHP es el acrónimo recursivo de Hipertext Preprocesor (inicialmente se llamó Personal Home Page); surgió en 1995, desarrollado por PHP Group¹⁴. Es un lenguaje de programación interpretado de alto nivel usado normalmente para la creación de páginas web dinámicas y ejecutadas en el servidor. Este lenguaje, además de ser gratuito e independiente de plataforma, posee una gran librería de funciones y mucha documentación. Entre sus principales características destacan su potencia, su alto rendimiento, su facilidad de aprendizaje y su escasez de consumo de recursos (31).

1.8 Framework

El investigador Jerome Laffose define un framework como: “... un conjunto de bibliotecas, herramientas y normas a seguir que ayudan a desarrollar aplicaciones” (32). Persiguen un objetivo común y es “... acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones” (33).

1.8.1 Frameworks para la capa de presentación

jQuery 1.9.1

jQuery es una biblioteca de JavaScript rápida y concisa que simplifica el recorrido de un documento HTML, de manejo de eventos, animaciones e interacciones Ajax¹⁵ para el desarrollo web rápido. Además de ser sencilla es una herramienta open source¹⁶, esto permite descargar sus librerías, modificarlas o personalizarlas de acuerdo a las necesidades del cliente. Estas características posibilitan que los contenidos se puedan mostrar y se interactúe con los mismos de forma dinámica, brindando la posibilidad de usar este framework para apoyar el cumplimiento de las funcionalidades necesarias (34).

En las Figuras 2 y 3 se muestra el uso de la tecnología jQuery.

¹⁴ Equipo de desarrollo que se encarga de mantener a diario actualizaciones disponibles para el lenguaje PHP.

¹⁵ Técnica de desarrollo web para crear aplicaciones interactivas.

¹⁶ En español código abierto.

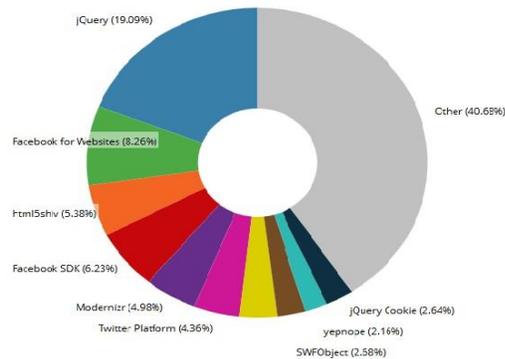


Figura 2: Estadísticas para sitios web usando la tecnología jQuery (35).

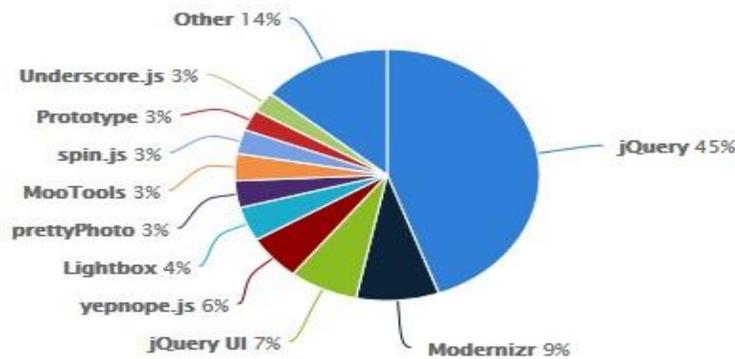


Figura 3: Cuota de mercado de las librerías JavaScript más populares en Internet (36).

Bootstrap 2.3.2

El framework Bootstrap surgió de la idea de hacer interfaces de aplicaciones web más amigables con el usuario. Logra definir una serie de etiquetas que posibilitan un maquetado de las interfaces de usuario a una mayor precisión. Entre sus principales características se destacan (37):

- Todos los estilos se encuentran en un único archivo y ha sido optimizado para lograr un buen rendimiento de las aplicaciones web.
- Las interfaces de usuario incluyen listado de grupos y paneles.
- Posee un grupo de componentes que lo hacen personalizable a cualquier ambiente de desarrollo de software.

- Presenta una facilidad de uso por parte de los diseñadores de interfaces de usuario, permitiendo un perfecto maquetado del contenido.

En la Figura 4 se muestra el uso de la tecnología Bootstrap.

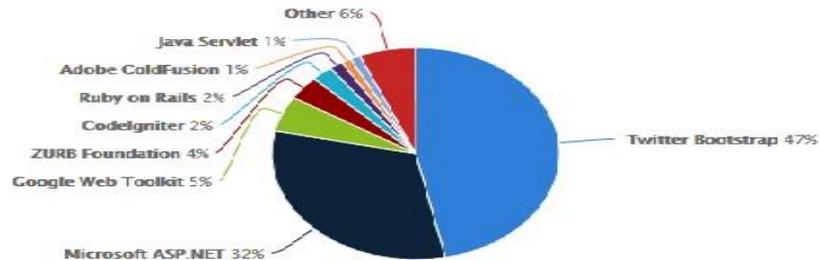


Figura 4 Cuota de mercado de los frameworks web más populares en Internet (38).

1.8.2 Frameworks para la capa de lógica de negocio

Symfony 2.5.0

Symfony es un framework de desarrollo web PHP, que separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación (39).

Es compatible con la mayoría de gestores de bases de datos, como MySQL¹⁷, PostgreSQL¹⁸, Oracle¹⁹ y Microsoft SQL Server²⁰.

Symfony fue diseñado para ajustarse a los siguientes requisitos (39):

- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de ORM²¹, permiten cambiar con facilidad de SGBD²² en cualquier fase del proyecto.

¹⁷ Sistema de gestión de base de datos relacional.

¹⁸ Sistema de gestión de base de datos orientado a objetos.

¹⁹ Sistema de gestión de base de datos desarrollado por Oracle Corporation.

²⁰ Sistema de gestión de base de datos creado por Microsoft.

²¹ Siglas de Mapeador de Objeto Relacional.

²² Siglas de Sistema Gestor de Base de Datos.

- Utiliza programación orientada a objetos y características como los espacios de nombres, de ahí que sea imprescindible PHP 5.4.
- Sencillo de usar en la mayoría de casos, aunque es preferible para el desarrollo de grandes aplicaciones Web que para pequeños proyectos.
- Aunque utiliza MVC (Modelo Vista Controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor ²³ y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
- Una potente línea de comandos que facilitan generación de código, lo cual contribuye a ahorrar tiempo de trabajo.

Además, permite el desarrollo automatizado (39):

- Permite la internacionalización para la traducción del texto de la interfaz, los datos y el contenido de localización.
- La presentación usa plantillas y capas que pueden ser construidas por diseñadores de HTML que no posean conocimientos del framework.
- Los formularios soportan la validación automática, lo cual asegura mejor calidad de los datos en las base de datos y una mejor experiencia para el usuario.
- El manejo de caché reduce el uso de banda ancha y la carga del servidor.

²³ Es un generador de código abierto escrito en PHP.

- La facilidad de soportar autenticación y credenciales facilita la creación de áreas restringidas y manejo de seguridad de los usuarios.
- El enrutamiento y las URLs²⁴ inteligentes hacen amigable las direcciones de las páginas de la aplicación.
- Las listas son más amigables, ya que permite la paginación, clasificación y filtraje automáticos.
- Los complementos proveen un alto nivel de extensibilidad.

1.8.3 Frameworks para la capa de acceso a datos

Un Mapeador de Objetos Relacionales (ORM, de sus siglas en inglés Object Relational Mapper), es una técnica de programación que permite convertir datos entre el sistema de datos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de la base de datos pasan a ser clases y los registros se pueden manejar con facilidad (40).

Doctrine 2.3

Doctrine es un ORM para PHP que proporciona persistencia transparente de los objetos desde el lenguaje PHP. Utiliza el patrón Data Mapper²⁵ en el núcleo del proyecto, logrando separar la lógica de negocio de la persistencia en un sistema de bases de datos relacional (40). Entre sus características se destacan:

- Soporte para las operaciones de Crear, Obtener, Actualizar y Borrar registros de la base de datos.
- Permite crear automáticamente el modelo de base de datos a implementar y presenta soporte para varios motores de bases de datos.
- Posibilita un trabajo más rápido y eficaz.
- Posee una amplia documentación lo que facilita la retroalimentación continua.

1.9 Servidor Web

Un servidor web es un servidor de archivos donde los clientes se dirigen a este mediante el protocolo HTTP²⁶ para obtener recursos. Cuando el servidor web recibe una petición HTTP, extrae simplemente de la petición el nombre del recurso solicitado, lo busca en el disco y lo envuelve dentro de una respuesta HTTP para

²⁴ En inglés Uniform Resource Locator, en español es Localizador Uniforme de Recursos.

²⁵ Es una capa de acceso de datos que realiza la transferencia bidireccional de datos entre una base de datos relacional y la capa de dominio.

²⁶ HTTP (Hypertext Transfer Protocol): protocolo usado en cada transacción de la World Wide Web.

transmitirlo al cliente. Un servidor no realiza ningún tratamiento en el recurso antes de transmitirlo al cliente. Por lo tanto puede transmitir de manera indiferente a un cliente una página HTML, una imagen, un archivo de sonido o incluso un archivo ejecutable (41).

Se puede decir además que un servidor web se mantiene a la espera de peticiones HTTP por parte de un cliente HTTP conocido como navegador. El cliente realiza una petición al servidor y este le responde con el contenido que el cliente solicita.

1.9.1 Apache 2.4.4

Apache es un servidor web HTTP de código abierto que su robustez, estabilidad, flexibilidad, seguridad, rapidez y eficiencia hacen que sea el más utilizado en el mundo (Ver Figuras 5 y 6). Entre sus características se destacan (42):

- Es multiplataforma.
- Proporciona un código fuente completo y viene con una licencia sin restricciones.
- Permite personalizar la respuesta ante los posibles errores y problemas que se puedan dar en el servidor.
- Posee un sistema de notificación de errores HTTP altamente configurable y de diseño modular.
- Constituye una tecnología gratuita de código abierto.
- Incentiva la retroalimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.

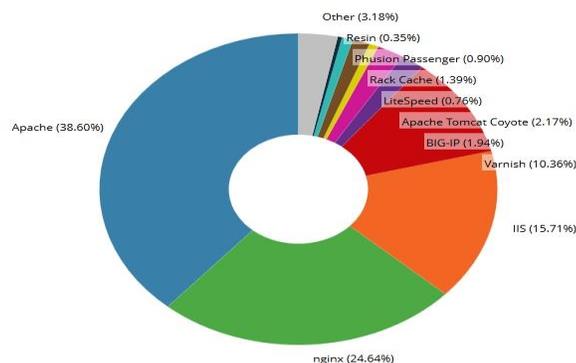


Figura 5: Estadísticas para sitios web usando servidor web (43).

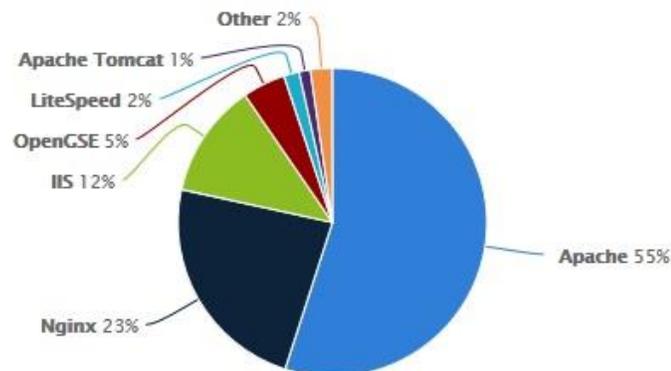


Figura 6: Cuota de mercado de los servidores web más populares en Internet (44).

1.10 Sistema gestor de base de datos

La bibliografía consultada permite definir para esta investigación un sistema gestor de base de datos (SGBD, en inglés DBMS: DataBase Management System) como: “... una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos” (45).

En otras palabras, un SGBD es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos vinculado con un conjunto de operaciones para manejar los mismos. Estos son una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo, ofrecen facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la agilidad, la seguridad y la integridad de los datos que contienen.

Entre sus principales características se destacan (45):

- **Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Consistencia.** En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra resguardada de usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Integridad.** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- **Respaldo y recuperación.** Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

1.10.1 PostgreSQL 9.2.4

PostgreSQL es un sistema de gestión de bases de datos objeto –relacional. Clasifica como software libre por su distribución bajo la licencia BSD²⁷, por lo que su código fuente y ficheros binarios pueden ser utilizados, modificados, redistribuidos y hasta incluidos en software con carácter no libre. Utiliza un modelo cliente –servidor y utiliza multiprocesos en vez de multihilos para garantizar que el sistema sea estable. Las características técnicas que posee lo hacen uno de los gestores más potentes y robustos del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en

²⁷ Licencia BSD (Berkeley Software Distribution), pertenece al grupo de licencias de software Libre pero tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público.

cuenta durante su desarrollo. Funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo al mismo tiempo al sistema (46). Algunas de las características más significativas de PostgreSQL son (46):

- PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, optimización de consultas, herencia y *arrays*²⁸ siendo de gran importancia para el trabajo con la aplicación.
- Soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- Incluye características avanzadas tales como las uniones (*joins*).
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- Posee claves ajenas también denominadas llaves ajenas o claves foráneas (*foreignkeys*).
- Incorpora disparadores (*triggers*): Un disparador o *trigger* se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica.

1.11 Entorno de desarrollo integrado

Un entorno de desarrollo integrado (IDE) es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un debug²⁹ y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación (47).

1.11.1 NetBeans 8.0

NetBeans es un sistema modular, basado en estándares entorno de desarrollo integrado, escrito en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto con todas las funciones escritas en el lenguaje de programación Java y una plataforma de aplicaciones cliente rico, que

²⁸ Un *array* es una colección ordenada de elementos de un mismo tipo de datos, agrupados de forma consecutiva en memoria.

²⁹ Programa usado para probar y eliminar errores de otros programas.

puede ser utilizado como un marco genérico para construir cualquier tipo de aplicación. Además permite a los desarrolladores escribir, compilar, depurar y ejecutar programas informáticos (48).

Conclusiones del capítulo

En el capítulo se realizó un análisis de los principales conceptos asociados a la investigación teniendo en cuenta la visión de diferentes autores, además de un estudio de los sistemas similares existentes en el mundo y en el país. Finalmente, se llegó a la conclusión de que los mismos no reúnen las condiciones necesarias para mejorar el proceso de gestión de la información de la GOE en la UCI, por lo que surge la necesidad de desarrollar un nuevo sistema que solucionará la problemática identificada.

Para el desarrollo del sistema propuesto se analizaron varias de las metodologías, tecnologías, herramientas y lenguajes existentes y se seleccionó: XP como metodología para guiar el proceso de desarrollo de software, UML como lenguaje de modelado y Visual Paradigm como herramienta CASE para realizar el modelado del software.

Tras determinarse que la propuesta de solución será una aplicación web, se emplearon tecnologías del lado del cliente y del servidor. En el lado del servidor el lenguaje seleccionado fue PHP con el framework Symfony, mientras que por el lado del cliente se seleccionó HTML para el maquetado, CSS para los estilos visuales a través de Bootstrap y como lenguaje de scripting JavaScript a través de la librería jQuery.

En el acceso a la base de datos se escogió el ORM Doctrine. Para el almacenamiento de la información se determinó a PostgreSQL como gestor de bases de datos. El servidor web a utilizar será Apache y por sus disímiles características que lo hacen un potente IDE, fue elegido NetBeans para desarrollar la propuesta de solución.

Capítulo 2: Construcción de la solución propuesta

2.1 Introducción

El presente capítulo se describe el proceso de construcción de la solución propuesta, definiendo de forma detallada los requisitos funcionales y no funcionales que debe cumplir la misma. También se describen los usuarios que intervienen en el sistema y las historias de usuarios que se utilizan durante el desarrollo. Además se realizan estimaciones de tiempo y esfuerzo para las iteraciones y para las historias de usuarios, así como las tarjetas Clase-Responsabilidad-Colaboración (CRC). Por último se exponen elementos referentes a la arquitectura del sistema.

2.2 Descripción de la solución propuesta

Para dar solución al problema planteado se propone la implementación de un sistema informático desarrollado mediante el framework Symfony, que permita mejorar el proceso de gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas. El sistema permitirá fácil acceso a los usuarios sin necesidad de emplear espacio en disco duro para su instalación y uso ya que es un sistema web. Proporcionará interfaces sencillas y fáciles de usar que le permiten al usuario interactuar de forma cómoda.

El proceso comienza una vez que el usuario se autentique y según los permisos que tenga asignado logrará consultar la información que desee. El desarrollo de este sistema facilitará contar con una herramienta capaz de optimizar su trabajo. Dentro del proceso los usuarios podrán desempeñar los siguientes roles: Administrador, Especialista de Control, Oficial de Guardia Superior y Directivos. El usuario con rol Administrador será el encargado de gestionar usuarios y las áreas. Luego el usuario con rol Oficial de Guardia Superior será capaz de generar un parte diario compuesto por afectaciones, violaciones, incidencias, hechos delictivos, visitas, extranjeros, acercamiento de extranjeros a la Universidad y la asistencia de la guardia. A continuación el usuario con rol Especialista de Control será el responsable de generar reportes de asistencias a diferentes niveles referentes al comportamiento de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas. Y por último el usuario con rol Directivos será el encargado de consultar estadísticas relacionadas con el comportamiento de la Guardia y exportar el resultado de la misma a formato PDF.

2.3 Usuarios relacionados con el sistema

Los usuarios del sistema son todas aquellas personas que de una forma u otra interactúan con el sistema y obtienen un resultado del valor de uno o varios procesos que se ejecutan en la misma. Se definieron los siguientes usuarios del sistema con sus responsabilidades.

Tabla 2: Usuarios del sistema.

Usuarios	Responsabilidad
Administrador	Tiene la responsabilidad de gestionar usuarios y las áreas.
Oficial de Guardia Superior	Encargado de controlar la asistencia de la Guardia Obrero Estudiantil. Además tiene la responsabilidad de generar un parte diario.
Especialista de Control	Tiene la responsabilidad de generar reportes de asistencias de la Guardia Obrero Estudiantil.
Directivos	Consultan estadísticas sobre el comportamiento de la Guardia Obrero Estudiantil.

2.4 Lista de reserva del producto

La lista de reserva del producto (LRP) es el artefacto que relaciona todos los requisitos funcionales del sistema ordenados en dependencia de su importancia (muy alta, alta, media o baja). Las siguientes tablas muestran la lista de reserva del producto con los requisitos funcionales y no funcionales identificados y sus respectivos niveles de prioridad.

Tabla 3: LRP-Requisitos Funcionales.

Código	Requisitos Funcionales (RF)	Prioridad
RF01	Autenticar usuario	Media
RF02	Crear usuario	Alta
RF03	Editar usuario	Alta
RF04	Eliminar usuario	Alta
RF05	Mostrar usuario	Media
RF06	Buscar usuario	Alta

RF07	Asignar rol	Alta
RF08	Crear área	Alta
RF09	Editar área	Alta
RF10	Eliminar área	Alta
RF11	Mostrar área	Media
RF12	Buscar área	Alta
RF13	Crear acercamiento de extranjeros	Alta
RF14	Editar acercamiento de extranjeros	Alta
RF15	Eliminar acercamiento de extranjeros	Alta
RF16	Mostrar acercamiento de extranjeros	Alta
RF17	Buscar acercamiento de extranjeros	Media
RF18	Crear incidencias	Alta
RF19	Editar incidencias	Alta
RF20	Eliminar incidencias	Alta
RF21	Mostrar incidencias	Alta
RF22	Buscar incidencias	Media
RF23	Crear hechos delictivos	Alta
RF24	Editar hechos delictivos	Alta
RF25	Eliminar hechos delictivos	Alta
RF26	Mostrar hechos delictivos	Alta
RF27	Buscar hechos delictivos	Media
RF28	Crear violaciones	Alta
RF29	Editar violaciones	Alta
RF30	Eliminar violaciones	Alta
RF31	Mostrar violaciones	Alta
RF32	Buscar violaciones	Media
RF33	Crear afectaciones	Alta
RF34	Editar afectaciones	Alta
RF35	Eliminar afectaciones	Alta
RF36	Mostrar afectaciones	Alta
RF37	Buscar afectaciones	Media
RF38	Crear visitas	Alta

RF39	Editar visitas	Alta
RF40	Eliminar visitas	Alta
RF41	Mostrar visitas	Alta
RF42	Buscar visitas	Media
RF43	Crear extranjeros	Alta
RF44	Editar extranjeros	Alta
RF45	Eliminar extranjeros	Alta
RF46	Mostrar extranjeros	Alta
RF47	Buscar extranjeros	Media
RF48	Crear guardia	Muy Alta
RF49	Editar guardia	Muy Alta
RF50	Eliminar guardia	Muy Alta
RF51	Mostrar guardia	Muy Alta
RF52	Buscar guardia	Media
RF53	Generar informe	Muy Alta
RF54	Generar reportes	Muy Alta
RF55	Consultar estadísticas	Media
RF56	Exportar a PDF	Baja

Tabla 4: LRP-Requisitos no Funcionales.

Código	Requisitos no Funcionales (RNF)
Usabilidad	
RnF01	Mantener coherencia y uniformidad en las estructuras y colores de todas las páginas para que el usuario no se desoriente en su navegación
RnF02	A los administradores del sistema se les dará un adiestramiento básico en el uso de la aplicación.
RnF03	El sistema estará disponible las 24 horas del día.
RnF04	El sistema propuesto podrá ser utilizado por personas que tengan conocimientos básicos de informática.
RnF05	El sistema debe satisfacer los requisitos funcionales declarados.
Eficiencia	
RnF06	El sistema debe soportar conexiones simultáneas de al menos 100 usuarios.

RnF07	La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos y la velocidad de la consultas de la base de datos.
RnF08	El sistema propuesto debe ser rápido y el tiempo de respuesta debe ser el mínimo posible, adecuado a la rapidez con que el cliente requiere la respuesta a su petición.
Seguridad	
RnF09	El sistema debe contar con un grupo de políticas de acceso en dependencia del nivel de autorización que presente un usuario determinado.
RnF10	La aplicación deberá estar disponible en todo momento para aquellas personas con acceso a la información.
RnF11	La seguridad se establecerá por roles que se le asignarán a los usuarios que interactúen con el sistema.
Soporte	
RnF12	Se necesita un servidor de bases de datos que soporte volúmenes de datos (PostgreSQL 9.*).
RnF13	Se necesita un servidor web (Apache 2.*).
RnF14	El sistema será multiplataforma (Linux o Windows).
RnF15	El sistema será de fácil instalación.
Restricciones de Software	
RnF16	Servidor de base de datos: Postgres 9.2 o Superior.
RnF17	Servidor de aplicaciones web: Apache 2.4 o Superior.
RnF18	Navegador web: Mozilla Firefox 36.0 o Superior, Google Chrome 38.0 o Superior, Safari 5.1 o Superior.
Restricciones de Hardware	
RnF19	En el cliente se requiere una máquina con 256 MB de RAM como mínimo.
RnF20	Todas las máquinas implicadas en la funcionalidad de la aplicación deben estar conectadas a la red.
RnF21	El disco duro no requiere gran cantidad de espacio.
RnF22	Procesador Intel® a 1 GHz. de velocidad de procesamiento o Superior.
Interfaz	
RnF23	La interfaz para el cliente debe ser sencilla y sin cúmulo de imágenes o elementos que lo distraigan de su objetivo.
RnF24	La interfaz debe ser amigable y fácil de usar, de manera que no sea una dificultad para los usuarios el trabajo con la misma.

2.5 Historias de usuarios

Las historias de usuario (HU) son la técnica utilizada en XP para especificar los requisitos del sistema a desarrollar (49). El cliente describió y priorizó sus necesidades mediante estas descripciones cortas y escritas sin terminología técnica. Se realizaron una por cada funcionalidad del sistema, se emplearon para hacer estimaciones de tiempo y para el plan de lanzamientos. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores pudieran implementarla en unas semanas.

A continuación se muestran algunas de las historias de usuario escritas para el sistema.

Tabla 5: HU-Autenticar usuario.

Historia de usuario	
Número: 1	Nombre: Autenticar usuario
Usuario: Administrador	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alto
Punto de Estimación: 0.5 semana	Iteración Asignada: 1
Descripción: <i>El usuario de la aplicación se autentica. Para esto se requiere:</i> <ul style="list-style-type: none"> • <i>Usuario (Obligatorio)</i> • <i>Contraseña (Obligatorio)</i> 	
Observación: <i>Si el usuario o la contraseña son incorrectos, tiene la posibilidad de introducirlos nuevamente.</i>	

Tabla 6: HU-Gestionar usuario.

Historia de usuario	
Número: 2	Nombre: Gestionar usuario
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 0.5 semana	Iteración Asignada: 1
Descripción: <i>El sistema permitirá que el Administrador pueda crear, editar, eliminar, mostrar y buscar un usuario en la base de datos. Para esto se requiere:</i> <ul style="list-style-type: none"> • <i>Nombre (Obligatorio)</i> • <i>Apellidos (Obligatorio)</i> 	

Observación: El usuario debe estar autenticado con el rol Administrador.

Tabla 7: HU-Gestionar área.

Historia de usuario	
Número: 4	Nombre: Gestionar área
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 1 semana	Iteración Asignada: 1
<p>Descripción: El sistema permitirá que el Administrador pueda crear, editar, eliminar, mostrar y buscar un área en la base de datos. Para esto se requiere:</p> <ul style="list-style-type: none"> • Nombre del área (Obligatorio) • Cantidad de postas (Obligatorio) • Categoría (Obligatorio) 	
Observación: El usuario tiene que estar autenticado con el rol Administrador.	

Tabla 8: HU-Gestionar hechos delictivos.

Historia de usuario	
Número: 7	Nombre: Gestionar hechos delictivos
Usuario: Oficial de Guardia Superior	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 1 semana	Iteración Asignada: 2
<p>Descripción: El sistema permitirá que el Oficial de Guardia Superior pueda crear, editar, eliminar, mostrar y buscar eliminar un hecho delictivo en la base de datos. Para esto se requiere:</p> <ul style="list-style-type: none"> • Tipo (Obligatorio) • Descripción (Obligatorio) • Hora (Obligatorio) 	
Observación: El usuario tiene que estar autenticado con el rol Oficial de Guardia Superior.	

Tabla 9: HU-Generar informe.

Historia de usuario	
Número: 13	Nombre: Generar informe
Usuario: Oficial de Guardia Superior	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 1.5 semana	Iteración Asignada: 3
<p>Descripción: El sistema permitirá que el Oficial de Guardia Superior pueda generar un informe diario en la base de datos. Para esto se requiere:</p> <ul style="list-style-type: none"> • Fecha (Obligatorio) 	

- Nombre del oficial (Obligatorio)
- Observación (Obligatorio)

Observación: El usuario tiene que estar autenticado con el rol Oficial de Guardia Superior.

2.6 Planificación

La planificación es una fase corta donde el cliente estableció la prioridad de cada historia de usuario y correspondientemente, el programador realizó una estimación del esfuerzo necesario de cada una de ellas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación y el resultado es un Plan de Entregas. Esta fase duró unos pocos días. Las estimaciones de esfuerzo asociado a la implementación de las historias las estableció el programador utilizando como medida el punto, lo que equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos (21).

2.6.1 Estimación de esfuerzo por historias de usuario

Para realizar un buen desarrollo del sistema propuesto, se realizó una estimación para cada una de las historias de usuario identificadas, llegando a los resultados que se muestran a continuación:

Tabla 10: Estimación de esfuerzo por HU.

Historia de Usuario	Estimación (semanas)
HU1. Autenticar usuario	0.5
HU2. Gestionar usuario	0.5
HU3. Asignar rol	0.5
HU4. Gestionar área	1
HU5. Gestionar acercamiento de extranjeros	1
HU6. Gestionar incidencias	1
HU7. Gestionar hechos delictivos	1
HU8. Gestionar violaciones	1
HU9. Gestionar afectaciones	1
HU10. Gestionar visitas	1
HU11. Gestionar extranjero	1
HU12. Gestionar guardia	1
HU13. Generar informe	1.5

HU14. Generar reportes	1.5
HU15. Consultar estadísticas	1
HU16. Exportar a PDF	0.5

2.6.2 Plan de entregas

El plan de entrega es un plan dividido en varios planes de iteración donde en la primera iteración se obtiene el sistema con toda su arquitectura, en dicho plan el usuario selecciona las HU en cada iteración y las pruebas funcionales de estas historias son validadas al final de la misma, además el cronograma de entregas establece cuáles HU serán agrupadas para conformar una entrega, y su orden. Este cronograma será el resultado de una reunión entre todos los actores del grupo de desarrollo incluyendo el cliente permitiendo alcanzar un mayor entendimiento en la implementación del sistema. Teniendo en cuenta las prioridades planteadas por el cliente y las estimaciones realizadas por los desarrolladores, se definió el siguiente Plan de Entregas:

Tabla 11: Plan de entregas.

Entregable	1ra entrega (1ra semana de Marzo)	2da entrega (1ra semana de Abril)	3ra entrega (2da semana de Mayo)
Sistema para la gestión de la información de la Guardia Obrera Estudiantil en la Universidad de las Ciencias Informáticas	Versión 0.1	Versión 0.2	Versión 1.0

Tabla 12: Funcionalidades disponibles por entrega del producto.

Historias de Usuario	1ra entrega	2da entrega	3ra entrega
HU1. Autenticar usuario	X		
HU2. Gestionar usuario	X		
HU3. Asignar rol	X		
HU4. Gestionar área	X		
HU5. Gestionar acercamiento de extranjeros	X		
HU6. Gestionar incidencias	X		
HU7. Gestionar hechos delictivos		X	

HU8. Generar violaciones		X	
HU9. Generar afectaciones		X	
HU10. Gestionar visitas		X	
HU11. Gestionar extranjero		X	
HU12. Gestionar guardia			X
HU13. Generar informe			X
HU14. Generar reportes			X
HU15. Consultar estadísticas			X
HU16. Exportar a PDF			X

2.6.3 Plan de Iteraciones

El plan de duración de las iteraciones se realiza luego de tener el estimado en días que demora implementar cada HU. Se tendrá en cuenta la prioridad que el cliente le asigna a cada historia y el nivel de complejidad que estas poseen.

Tabla 13: Plan de iteraciones.

Iteraciones	Historias de usuario	Duración
Iteración 1	HU1. Autenticar usuario HU2. Gestionar usuario HU3. Asignar rol HU4. Gestionar área HU5. Gestionar acercamiento de extranjeros HU6. Gestionar incidencias	4.5 semanas
Iteración 2	HU7. Gestionar hechos delictivos HU8. Generar violaciones HU9. Generar afectaciones HU10. Gestionar visitas HU11. Gestionar extranjero	5 semanas
Iteración 3	HU12. Gestionar guardia HU13. Generar informe HU14. Generar reportes	5.5 semanas

	HU15. Consultar estadísticas HU16. Exportar a PDF	
--	--	--

2.7 Diseño del sistema

La metodología seleccionada no requiere la representación del sistema mediante diagramas de clases utilizando notación UML. En su lugar se usan otras técnicas como las tarjetas CRC como una extensión informal a UML. No obstante el uso de estos diagramas, puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante. El uso de las tarjetas CRC permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica.

2.7.1 Tarjetas Clase-Responsabilidad-Colaboración (CRC)

Las tarjetas CRC son en la práctica pequeñas tarjetas que se elaboran para ser mostradas al cliente, de manera que se pueda llegar a un acuerdo sobre la validez de las abstracciones propuestas, lo que ayuda al equipo durante el diseño e implementación del sistema. Estas constituyen documentación adicional que será adjuntada a las HU.

Las tarjetas CRC trabajan con la técnica de modelado basada en objetos, representando cada tarjeta CRC a un objeto, identificando las clases y sus responsabilidades. Las tarjetas están compuestas por el nombre de la clase, responsabilidades y colaboraciones (49). Para la solución se diseñaron las siguientes tarjetas CRC:

Tabla 14: Tarjeta CRC-Usuario.

Usuario	
Responsabilidad	Colaboración
Gestionar usuario	
Asignar rol	
Autenticar usuario	

Tabla 15: Tarjeta CRC-Área.

Área	
Responsabilidad	Colaboración

Gestionar área	
----------------	--

Tabla 16: Tarjeta CRC-Hechos delictivos.

Hechos delictivos	
Responsabilidad	Colaboración
Gestionar hechos delictivos	Guardia

Tabla 17: Tarjeta CRC-Parte diario.

Parte diario	
Responsabilidad	Colaboración
Generar informes	
Generar reportes	
Exportar a PDF	

2.7.2 Arquitectura

La arquitectura del software es la organización fundamental de un sistema, formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución (50).

Para definir la arquitectura que se empleará en la presente solución, es necesaria la selección de los patrones arquitectónicos y de diseño más adecuados.

2.7.2.1 Patrones de arquitectura

Un patrón arquitectónico constituye una descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución. Expresa el esquema de organización estructural fundamental para sistemas de software. Provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos. Es una plantilla para arquitecturas de software concretas, que especifica las propiedades estructurales de una aplicación (51).

El framework seleccionado para el desarrollo del sistema a implementar, está totalmente basado en la arquitectura Modelo Vista Controlador (MVC). Esta arquitectura permite dividir la aplicación en tres grandes capas (52):

- **Vista:** Es lo que utilizan los usuarios para interactuar con la aplicación (los gestores de plantillas pertenecen a esta capa).
- **Modelado:** Es la capa del modelo define la lógica de negocio (la base de datos pertenece a esta capa).
- **Controlador:** Es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario.

En Symfony, el acceso y la modificación de los datos que se almacenan en la base de datos, se realiza mediante objetos. La implementación que realiza Symfony de la arquitectura incluye clases como:

- **Controller:** Es la clase del controlador y se encarga de decodificar la petición y transferirla a la acción correspondiente.
- **Request:** Guarda todos los elementos que integran la petición (parámetros, cookies, cabeceras.)
- **Response:** Posee las cabeceras de la respuesta y los contenidos. El contenido de este objeto se convierte en la respuesta HTML que se remite al usuario.

El cliente envía una señal llamada request o petición, ésta es interceptada por el controlador quien realiza las validaciones necesarias, procesamiento de dichos datos y lógica de negocio asociadas a esa petición del cliente. El controlador envía datos al modelado, por ejemplo para ser guardados en una base de datos y/o los obtiene dependiendo de la solicitud del usuario para finalmente enviarlos a la vista a fin de ser mostrador nuevamente al cliente a través de un response o respuesta. (Ver Figura 7)

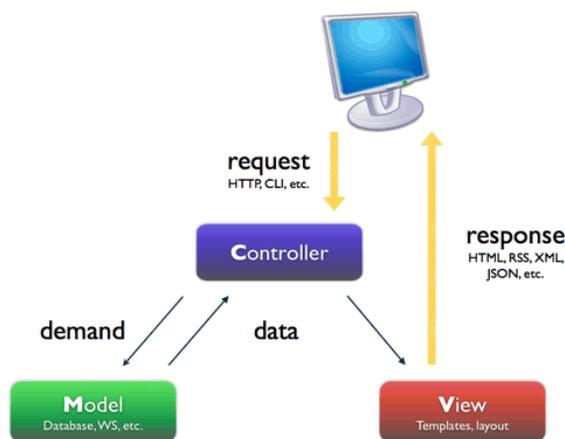


Figura 7: Funcionamiento básico del patrón MVC en Symfony.

2.7.2.2 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Estos patrones identifican clases, instancias, roles, colaboraciones y la distribución de responsabilidades (53). Los patrones de diseño utilizados para el desarrollo de la solución propuesta son los que se presentan a continuación.

Patrones GRASP³⁰

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable (53).

De los patrones GRASP existentes, se emplearon en el diseño de la solución los siguientes:

Experto: Este patrón plantea la asignación de responsabilidades será para la clase que contiene la información necesaria para cumplir la responsabilidad. Permite conservar el encapsulamiento ya que los objetos se valen de su propia información para hacer lo que se les pide, lo que provee un bajo nivel de acoplamiento. Promueve clases sencillas y cohesivas que son más fáciles de mantener y comprender (53). Este patrón es utilizado en la capa de abstracción del modelo de datos a través del ORM Doctrine. Las clases generadas poseen un grupo de funcionalidades que facilitan el acceso y la manipulación de los datos de las entidades persistentes en la base de datos (54). A continuación se presenta un ejemplo con la clase entidad “Area.php”, implementada en la propuesta:

³⁰ Del inglés General Responsibility Assignment Software Patterns, en español Patrones Generales de Software para Asignar Responsabilidades.

```
<?php

namespace Proyecto\GuardiaBundle\Entity;

use Doctrine\ORM\Mapping as ORM;

/**
 * Area
 *
 * @ORM\Table(name="area")
 * @ORM\Entity
 */
class Area
{
    /**
     * @var integer
     *
     * @ORM\Column(name="id", type="integer", nullable=false)
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="SEQUENCE")
     * @ORM\SequenceGenerator(sequenceName="area_id_seq", allocationSize=1, initialValue=1)
     */
}
```

Figura 8: Declaración de la clase Área.

Creador: Este patrón resuelve el problema de asignar responsabilidades relacionadas con la creación de objetos (53). Es utilizado en los controladores, en ellos se encuentran las acciones definidas para el sistema. En la implementación de las acciones se crean los objetos de las clases que presentan las entidades evidenciando que las clases controladoras son creadoras de dichas entidades. (Ver Figura 9)

```
public function createAction(Request $request)
{
    $entity = new Area();
    $form = $this->createForm($entity);
    $form->handleRequest($request);

    if ($form->isValid()) {
        $em = $this->getDoctrine()->getManager();
        $em->persist($entity);
        $em->flush();

        $this->get('session')->getFlashBag()->add(
            'notice', FlashMessageManager::CreateFlash
        );

        return $this->redirect($this->generateUrl('area_show', array('id' => $entity->getId())));
    }

    return $this->render('GuardiaBundle:Area:new.html.twig', array(
        'entity' => $entity,
        'form' => $form->createView(),
    ));
}
```

Figura 9: Función donde se crea un objeto de la clase Área.

Bajo Acoplamiento: Consiste en asignar una responsabilidad para mantener bajo acoplamiento. Es una medida de la fuerza con que una clase está conectada a otras clases. Este patrón da soporte a una mínima dependencia entre clases y a un aumento de la reutilización. De este modo una modificación en alguna de

las clases tiene la mínima repercusión posible en el resto de las clases (53). Esto se evidencia en la clase “AreaController.php” que hereda de la clase “Controller.php”.

Alta cohesión: Consiste en asignar una responsabilidad de modo que la cohesión siga siendo alta. En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme (53). En la solución se evidencia en la implementación de las clases entidades.

Controlador: Consiste en un objeto que no pertenece a la interfaz de usuario, responsable de recibir o manejar un evento del sistema, define el método para la operación del sistema (53). En Symfony se evidencia en las clases que forman la capa Controlador del patrón arquitectónico MVC, como son: “AreaController.php” y “GuardiaController.php”.

Patrones GOF³¹

Los patrones de diseño GOF se caracterizan por indicar resoluciones técnicas basadas en Programación Orientada a Objetos (POO). En ocasiones tienen más utilidad con algunos lenguajes de programación y en otras son aplicables a cualquier lenguaje, además favorecen la reutilización de código pues ayudan a construir un software basado en la reutilización.

De los patrones GOF existentes, se emplearon en el diseño de la solución los siguientes:

Decorator: Es un patrón que añade una funcionalidad a una clase dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades (54). En la solución propuesta se evidencia este patrón con la vista “administrador.html.twig”.



³¹ Del inglés Gang of Four, en español Banda de los Cuatro.

Figura 10: Patrón Decorator en Symfony.

2.7.3 Modelo de datos

Un modelo es un conjunto de herramientas conceptuales para describir datos, sus relaciones, su significado y sus restricciones de consistencia. A continuación se muestra una propuesta del modelo de datos del Sistema para la gestión de la información de la GOE en la UCI.

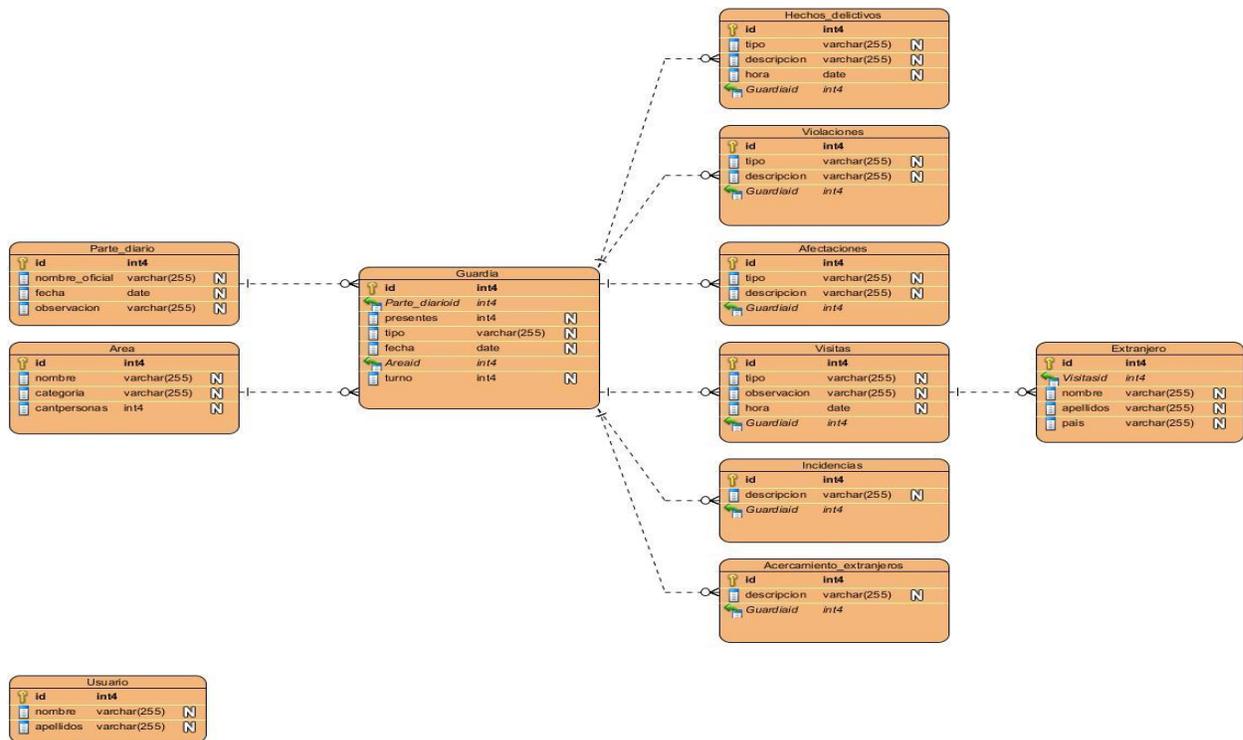


Figura 11: Modelo de datos.

Conclusiones del capítulo

En este capítulo se describió la propuesta de solución determinando las actividades que serán automatizadas dentro del proceso de gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas. La metodología de desarrollo seleccionada para guiar el proceso de software, posibilitó la descripción de los 4 tipos de usuarios que interactúan con el sistema, garantizando la seguridad de acceso al contenido. Conjuntamente con el cliente se definieron 3 iteraciones en un tiempo de 15 semanas, que abarcan un total de 16 HU (13 con prioridad alta, 2 media y 1 baja), facilitando la

comunicación del programador con el mismo a la hora de implementarlas, teniendo en cuenta, según el plan de iteraciones, el orden de su implementación. Se obtuvo un plan de entregas del sistema, determinando un cronograma en conjunto con el cliente. Además se elaboraron 11 tarjetas CRC y se ejemplificó la aplicación de los patrones de diseño.

Capítulo 3: Implementación y pruebas del sistema

3.1 Introducción

En este capítulo se exponen aspectos vinculados a las fases de implementación y prueba de la solución. La metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo, al culminar cada iteración, un producto funcional que debe ser probado y mostrado al cliente para incrementar la visión de los desarrolladores con la opinión de éste. Además se realizan todas las pruebas unitarias y las pruebas de aceptación al sistema, con el objetivo de garantizar el cumplimiento de las funcionalidades propuestas y garantizar la calidad del producto.

3.2 Estándares de codificación

Un estándar de codificación comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, se recomienda establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada. Usar técnicas de codificación sólidas es de gran importancia para la calidad del software y para obtener un buen rendimiento (55).

A continuación se muestran algunas de estas convenciones tenidas en cuenta para la implementación de la aplicación.

1. El número de caracteres por línea deben ser de 80 columnas aunque también está aceptado que sean hasta 120.
2. Los métodos y funciones deben ser escritos utilizando la técnica camelCase.
3. Los comentarios multilíneas se escriben comenzando con los caracteres `/*` y terminando con `*/`, los comentarios de una sola línea comienzan con los caracteres `//`.
4. Todas las clases controladoras comienzan con letra mayúsculas y terminan con la palabra Controller. En el caso de que las clases controladoras contengan más de una palabra, las iniciales de cada palabra comienzan con mayúsculas y el resto con minúscula, ejemplo de estas clases se tiene la siguiente: `AreaController ()`.

5. Todos los formularios comiencen con letra mayúscula y terminen con la palabra "Type", ejemplo de ello: AreaType ().
6. Las variables deben ser explícitas, aunque se pueden usar abreviaturas siempre y cuando no violen este principio.
7. Usa namespaces para todas clases.

3.3 Implementación

Durante esta fase se procede a implementar las HU relacionadas con las tres iteraciones descritas. Este proceso está guiado por el plan de iteraciones y descrito con la utilización de las Tareas de Ingeniería. En la metodología XP se define como Tarea de Ingeniería al conjunto de acciones asociadas a cada HU. Estas permiten organizar el proceso de implementación además de posibilitar que sea conocido el grado de complejidad de cada historia de usuario teniendo en cuenta la cantidad de tareas asociadas a ella. En cada una de las tareas se da a conocer el programador asignado, así como el tiempo necesario para su realización, lo que facilita la estimación del tiempo que abarcará cada historia de usuario en implementarse, de acuerdo con su complejidad. Esta fase se basa en la implementación de código, atendiendo a los estándares de codificación definidos con anterioridad ya que facilita la comprensión de cualquier miembro del equipo de desarrollo (20).

3.3.1 Tareas de Ingeniería

Las tareas de ingeniería son actividades que los programadores conocen que el sistema debe hacer. Deben ser estimables, y poder ser implementadas entre uno y tres días ideales. La mayoría de las tareas de programación se derivan directamente de las historias de usuario. Una historia de usuario puede tener una o varias tareas de ingeniería, en dependencia de la funcionalidad a desarrollar (20).

A continuación se detallan las tareas de ingeniería que se realizarán por cada iteración durante la implementación de la solución.

3.3.1.1 Iteración 1

Esta iteración tiene como objetivo dar cumplimiento a las HU 1, 2, 3, 4, 5 y 6 las cuales comprenden autenticar usuario, gestionar usuario, rol, área, acercamiento de extranjeros e incidencias respectivamente.

Tabla 18: Historias de Usuario implementadas en la 1ra Iteración.

Historias de Usuarios	Estimación	Real
HU1. Autenticar usuario	0.5	0.5
HU2. Gestionar usuario	0.5	0.5
HU3. Asignar rol	0.5	0.5
HU4. Gestionar área	1	1
HU5. Gestionar acercamiento de extranjeros	1	1
HU6. Gestionar incidencias	1	1
Total	4.5	4.5

Tabla 19: Tarea 1 de la HU Autenticar usuario.

Tarea de ingeniería	
Número: 1	Número de HU: 1
Nombre de la tarea: <i>Autenticar usuario</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.5
Fecha inicio: 1 de Febrero de 2015	Fecha fin: 3 de Febrero de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementa el diseño de la interfaz de autenticación, donde el usuario debe introducir un usuario y una contraseña.</i>	

Tabla 20: Tarea 2 de la HU Gestionar usuario.

Tarea de ingeniería	
Número: 2	Número de HU: 2
Nombre de la tarea: <i>Crear usuario</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.1
Fecha inicio: 3 de Febrero de 2015	Fecha fin: 4 de Febrero de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten crear un usuario en la base de datos.</i>	

Tabla 21: Tarea 3 de la HU Gestionar usuario.

Tarea de ingeniería	
Número: 3	Número de HU: 2
Nombre de la tarea: <i>Editar usuario</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.1
Fecha inicio: 4 de Febrero de 2015	Fecha fin: 4 de Febrero de 2015

Programador responsable: Raúl Rosabal Sánchez
Descripción: Se implementan las funcionalidades que permiten editar un usuario en la base de datos.

Tabla 22: Tarea 4 de la HU Gestionar usuario.

Tarea de ingeniería	
Número: 4	Número de HU: 2
Nombre de la tarea: Eliminar usuario	
Tipo de tarea: Desarrollo	Estimación: 0.1
Fecha inicio: 5 de Febrero de 2015	Fecha fin: 5 de Febrero de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten eliminar un usuario en la base de datos.	

Tabla 23: Tarea 5 de la HU Gestionar usuario.

Tarea de ingeniería	
Número: 5	Número de HU: 2
Nombre de la tarea: Mostrar usuario	
Tipo de tarea: Desarrollo	Estimación: 0.1
Fecha inicio: 5 de Febrero de 2015	Fecha fin: 6 de Febrero de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten mostrar un usuario en la base de datos.	

Tabla 24: Tarea 6 de la HU Gestionar usuario.

Tarea de ingeniería	
Número: 6	Número de HU: 2
Nombre de la tarea: Buscar usuario	
Tipo de tarea: Desarrollo	Estimación: 0.1
Fecha inicio: 6 de Febrero de 2015	Fecha fin: 7 de Febrero de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten buscar un usuario en la base de datos.	

Tabla 25: Tarea 8 de la HU Gestionar área.

Tarea de ingeniería	
Número: 8	Número de HU: 4
Nombre de la tarea: Crear área	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 11 de Febrero de 2015	Fecha fin: 12 de Febrero de 2015
Programador responsable: Raúl Rosabal Sánchez	

Descripción: Se implementan las funcionalidades que permiten crear un área en la base de datos.

Tabla 26: Tarea 9 de la HU Gestionar área.

Tarea de ingeniería	
Número: 9	Número de HU: 4
Nombre de la tarea: Editar área	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 12 de Febrero de 2015	Fecha fin: 13 de Febrero de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten editar un área en la base de datos.	

Tabla 27: Tarea 10 de la HU Gestionar área.

Tarea de ingeniería	
Número: 10	Número de HU: 4
Nombre de la tarea: Eliminar área	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 14 de Febrero de 2015	Fecha fin: 15 de Febrero de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten eliminar un área en la base de datos.	

Tabla 28: Tarea 11 de la HU Gestionar área.

Tarea de ingeniería	
Número: 11	Número de HU: 4
Nombre de la tarea: Mostrar área	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 15 de Febrero de 2015	Fecha fin: 16 de Febrero de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten mostrar un área en la base de datos.	

Tabla 29: Tarea 12 de la HU Gestionar área.

Tarea de ingeniería	
Número: 12	Número de HU: 4
Nombre de la tarea: Buscar área	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 16 de Febrero de 2015	Fecha fin: 17 de Febrero de 2015
Programador responsable: Raúl Rosabal Sánchez	

Descripción: Se implementan las funcionalidades que permiten buscar un área en la base de datos.

3.3.1.2 Iteración 2

Esta iteración tiene como objetivo dar cumplimiento a las HU 7, 8, 9, 10 y 11 las cuales comprenden gestionar hechos delictivos, violaciones, afectaciones, vistas y extranjeros respectivamente.

Tabla 30: Historias de Usuario implementadas en la 2da Iteración.

Historias de Usuarios	Estimación	Real
HU7. Gestionar hechos delictivos	1	1
HU8. Generar violaciones	1	1
HU9. Generar afectaciones	1	1
HU10. Gestionar visitas	1	1
HU11. Gestionar extranjero	1	1
Total	5	5

Tabla 31: Tarea 23 de la HU Gestionar hechos delictivos.

Tarea de ingeniería	
Número: 23	Número de HU: 7
Nombre de la tarea: <i>Crear hechos delictivos</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: <i>1 de Marzo de 2015</i>	Fecha fin: <i>2 de Marzo de 2015</i>
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten crear un hecho delictivo en la base de datos.</i>	

Tabla 32: Tarea 24 de la HU Gestionar hechos delictivos.

Tarea de ingeniería	
Número: 24	Número de HU: 7
Nombre de la tarea: <i>Editar hechos delictivos</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: <i>2 de Marzo de 2015</i>	Fecha fin: <i>3 de Marzo de 2015</i>
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten editar un hecho delictivo en la base de datos.</i>	

Tabla 33: Tarea 25 de la HU Gestionar hechos delictivos.

Tarea de ingeniería	
Número: 25	Número de HU: 7
Nombre de la tarea: <i>Eliminar hechos delictivos</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 4 de Marzo de 2015	Fecha fin: 5 de Marzo de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten eliminar un hecho delictivo en la base de datos.</i>	

Tabla 34: Tarea 26 de la HU Gestionar hechos delictivos.

Tarea de ingeniería	
Número: 26	Número de HU: 7
Nombre de la tarea: <i>Mostrar hechos delictivos</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 5 de Marzo de 2015	Fecha fin: 6 de Marzo de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten mostrar un hecho delictivo en la base de datos.</i>	

Tabla 35: Tarea 27 de la HU Gestionar hechos delictivos.

Tarea de ingeniería	
Número: 27	Número de HU: 7
Nombre de la tarea: <i>Buscar hechos delictivos</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 6 de Marzo de 2015	Fecha fin: 7 de Marzo de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten buscar un hecho delictivo en la base de datos.</i>	

3.3.1.3 Iteración 3

Esta iteración tiene como objetivo dar cumplimiento a las HU 12, 13, 14, 15 y 16 las cuales comprenden gestionar guardia, generar informes, generar reportes, consultar estadísticas y exportar a PDF respectivamente.

Tabla 36: Historias de Usuario implementadas en la 3ra Iteración.

Historias de Usuarios	Estimación	Real
HU12. Gestionar guardia	1	1
HU13. Generar informe	1.5	1.5
HU14. Generar reportes	1.5	1.5
HU15. Consultar estadísticas	1	1
HU16. Exportar a PDF	0.5	0.5
Total	5.5	5.5

Tabla 37: Tarea 48 de la HU Gestionar guardia.

Tarea de ingeniería	
Número: 48	Número de HU: 12
Nombre de la tarea: <i>Crear guardia</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 4 de Abril de 2015	Fecha fin: 5 de Abril de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten crear una guardia en la base de datos.</i>	

Tabla 38: Tarea 49 de la HU Gestionar guardia.

Tarea de ingeniería	
Número: 49	Número de HU: 12
Nombre de la tarea: <i>Editar guardia</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 5 de Abril de 2015	Fecha fin: 6 de Abril de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten editar una guardia en la base de datos.</i>	

Tabla 39: Tarea 50 de la HU Gestionar guardia.

Tarea de ingeniería	
Número: 50	Número de HU: 12
Nombre de la tarea: <i>Eliminar guardia</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 7 de Abril de 2015	Fecha fin: 8 de Abril de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	

Descripción: Se implementan las funcionalidades que permiten eliminar una guardia en la base de datos.

Tabla 40: Tarea 51 de la HU Gestionar guardia.

Tarea de ingeniería	
Número: 51	Número de HU: 12
Nombre de la tarea: <i>Mostrar guardia</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 8 de Abril de 2015	Fecha fin: 9 de Abril de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: Se implementan las funcionalidades que permiten mostrar una guardia en la base de datos.	

Tabla 41: Tarea 52 de la HU Gestionar guardia.

Tarea de ingeniería	
Número: 52	Número de HU: 12
Nombre de la tarea: <i>Buscar guardia</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 9 de Abril de 2015	Fecha fin: 10 de Abril de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: Se implementan las funcionalidades que permiten buscar una guardia en la base de datos.	

Tabla 42: Tarea 53 de la HU Generar informe.

Tarea de ingeniería	
Número: 53	Número de HU: 13
Nombre de la tarea: <i>Generar informe</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 1.5
Fecha inicio: 11 de Abril de 2015	Fecha fin: 21 de Abril de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: Se implementan las funcionalidades que permiten generar un informe diario sobre la asistencia de la GOE.	

Tabla 43: Tarea 54 de la HU Generar reportes.

Tarea de ingeniería	
Número: 54	Número de HU: 14
Nombre de la tarea: <i>Generar reportes</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 1.5
Fecha inicio: 21 de Abril de 2015	Fecha fin: 1 de Mayo de 2015

Programador responsable: Raúl Rosabal Sánchez
Descripción: Se implementan las funcionalidades que permiten generar reportes sobre la GOE.

Tabla 44: Tarea 55 de la HU Consultar estadísticas.

Tarea de ingeniería	
Número: 55	Número de HU: 15
Nombre de la tarea: Consultar estadísticas	
Tipo de tarea: Desarrollo	Estimación: 1
Fecha inicio: 1 de Mayo de 2015	Fecha fin: 7 de Mayo de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten consultar estadísticas referentes al comportamiento de la GOE.	

Tabla 45: Tarea 56 de la HU Exportar a PDF.

Tarea de ingeniería	
Número: 56	Número de HU: 16
Nombre de la tarea: Exportar a PDF	
Tipo de tarea: Desarrollo	Estimación: 0.5
Fecha inicio: 7 de Mayo de 2015	Fecha fin: 10 de Mayo de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten exportar datos referentes al comportamiento de la GOE a formato PDF.	

3.4 Pruebas

Unos de los pasos más importantes de la metodología XP es el proceso de pruebas, el cual anima a los desarrolladores a probar constantemente la aplicación tanto como sea posible. Mediante esta filosofía se reduce el número de errores no detectados, así como el tiempo entre la introducción de éste en el sistema y su detección. Todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores, a la hora de introducir cambios o modificaciones.

La metodología XP divide las pruebas en dos grupos: pruebas unitarias; desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación; destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente (56).

3.4.1 Pruebas unitarias

Las pruebas unitarias o pruebas consisten en comprobaciones (manuales o automatizadas) desarrolladas por los programadores. Las cuales se realizan para verificar el comportamiento de cada uno de los componentes ya sean módulos o historias de usuario de forma independiente. Este tipo de prueba permite producir un código de mayor calidad, detectar errores cuando se programan nuevas funcionalidades o se realizan cambios en el código. Sirven como pequeña fuente de documentación sobre qué es lo que se espera que haga el código y obliga a los desarrolladores a escribir el código en pequeñas porciones con el fin de que puedan ser probadas independientemente (56).

Para realizar las pruebas unitarias se utilizó el framework PHPUnit³², este permite realizar las pruebas pertinentes al código, verificando que el funcionamiento de las aplicaciones PHP es el deseado y en su caso encontrando bugs³³ y errores que una vez solucionados mejorarán la calidad del sistema. PHPUnit es sencillo de programar y fácil de utilizar. Es muy recomendado por los desarrolladores debido a que presenta características comunes con otros frameworks de testeo.

A continuación se describen algunos métodos que provee PHPUnit que fueron utilizados durante la ejecución de las pruebas.

Tabla 46: Métodos del framework PHPUnit para las pruebas unitarias (52).

Método	Descripción
getPropertyPath()	Devuelve el nombre de la propiedad que ha producido el error de validación.
getMessage()	Devuelve el mensaje de error completo que se mostraría al usuario.
getInvalidValue()	Devuelve el valor de la propiedad que ha producido el error de validación.
assertEquals(\$valor_esperado, \$valor_obtenido, \$mensaje)	Comprueba que los dos primeros argumentos que se le pasan son iguales.
assertRegExp('exp', \$valor)	Comprueba si el valor obtenido cumple con la expresión regular indicada como primer parámetro.
assertType()	Comprueba que una variable es de un cierto tipo.
assertContains()	Comprueba que la entrada contiene un valor específico.
setUp()	Antes de ejecutar los tests, PHPUnit busca un método llamado setUp () dentro de la clase. Si existe, lo ejecuta antes que cualquier test, por lo que es el lugar ideal para inicializar y preparar cualquier elemento que necesiten los tests.

³² Es un entorno para realizar pruebas unitarias en el lenguaje de programación PHP.

³³ Error de software.

getMessageTemplate()	Devuelve la plantilla utilizada para generar el mensaje de error. Puede contener variables de Twig como por ejemplo <code>{{cant}}</code> .
getMessageParameters()	Devuelve un array con los parámetros que se pasan a la plantilla para generar el mensaje de error que finalmente se muestra al usuario. Ejemplo: array (' <code>{{limit}}</code> ' => 30).
getRoot()	Devuelve el objeto que se está validando, por lo que proporciona acceso a todas las propiedades del objeto original.

3.4.2 Pruebas de aceptación

Las pruebas de aceptación tienen mayor importancia que las pruebas unitarias, dado que significa la satisfacción del cliente con el producto desarrollado, por consiguiente el cliente es la persona indicada para diseñar las pruebas a ejecutar.

Las pruebas de aceptación se elaboran a lo largo de la iteración, en paralelo con el desarrollo del sistema, y adaptándose a los cambios que el sistema sufra. Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación (56). A continuación aparecen los casos de pruebas de aceptación referentes a la propuesta de solución:

Tabla 47: Caso de prueba de aceptación #1.

Caso de prueba de aceptación	
Código: HU1_CP1	Historia de usuario: 1
Nombre: Autenticar usuario	
Descripción: Prueba para la funcionalidad autenticar usuario.	
Condiciones de ejecución: El usuario debe llenar correctamente los campos requeridos.	
Entrada/ Pasos de ejecución: El usuario procede a la inserción de los datos necesarios especificando su usuario y contraseña y presiona sobre el botón “Iniciar sesión”.	
Resultado esperado: El usuario se autentica correctamente en el sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 48: Caso de prueba de aceptación #2.

Caso de prueba de aceptación	
Código: HU2_CP1	Historia de usuario: 2
Nombre: Crear usuario	
Descripción: Prueba para la funcionalidad crear usuario.	

Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Administrador para tener acceso a la funcionalidad.</i>
Entrada/ Pasos de ejecución: <i>El usuario con rol Administrador introduce los datos requeridos para ser insertado en la base de datos. Para insertar los datos debe ir al menú y seleccionar la opción "Usuario". En la ventana que se va a mostrar debe presionar en el botón "Nuevo". Una vez introducidos los datos se presiona en el botón "Crear" y seguido se le muestra el usuario añadido.</i>
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>
Evaluación de la prueba: <i>Prueba satisfactoria.</i>

Tabla 49: Caso de prueba de aceptación #3.

Caso de prueba de aceptación	
Código: HU2_CP2	Historia de usuario: 2
Nombre: <i>Editar usuario</i>	
Descripción: <i>Prueba para la funcionalidad editar usuario.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Administrador y además debe haber algún usuario insertado en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>El usuario con rol Administrador introduce los datos requeridos para ser modificados en la base de datos. Para editar los datos debe ir al menú y seleccionar la opción "Usuario". En la ventana que se va a mostrar debe presionar la opción que indica editar un usuario. Una vez modificados los datos se presiona en el botón "Actualizar" y seguido se le muestra el usuario modificado.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 50: Caso de prueba de aceptación #4.

Caso de prueba de aceptación	
Código: HU2_CP3	Historia de usuario: 2
Nombre: <i>Eliminar usuario</i>	
Descripción: <i>Prueba para la funcionalidad eliminar usuario.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Administrador y además debe haber algún usuario insertado en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para eliminar un usuario debe ir al menú y seleccionar la opción "Usuario". En la ventana que se va a mostrar debe presionar la opción que indica eliminar un usuario, y se le pregunta al Administrador si está seguro que desea eliminar el usuario.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 51: Caso de prueba de aceptación #5.

Caso de prueba de aceptación	
Código: HU2_CP4	Historia de usuario: 2
Nombre: <i>Mostrar usuario</i>	

Descripción: Prueba para la funcionalidad mostrar usuario.
Condiciones de ejecución: El usuario debe estar autenticado con el rol Administrador y además debe haber algún usuario insertado en la base de datos.
Entrada/ Pasos de ejecución: Para mostrar un usuario debe ir al menú y seleccionar la opción "Usuario". En la ventana que se va a mostrar debe presionar la opción que indica mostrar un usuario.
Resultado esperado: Se mostrarán los datos del usuario seleccionado.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 52: Caso de prueba de aceptación #6.

Caso de prueba de aceptación	
Código: HU4_CP1	Historia de usuario: 4
Nombre: Crear área	
Descripción: Prueba para la funcionalidad crear área.	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Administrador para tener acceso a la funcionalidad.	
Entrada/ Pasos de ejecución: El usuario con rol Administrador introduce los datos requeridos para ser insertado en la base de datos. Para insertar los datos debe ir al menú y seleccionar la opción "Área". En la ventana que se va a mostrar debe presionar en el botón "Nuevo". Una vez introducidos los datos se presiona en el botón "Crear" y seguido se le muestra el área añadida.	
Resultado esperado: Mostrar un mensaje indicando el éxito de la operación.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 53: Caso de prueba de aceptación #7.

Caso de prueba de aceptación	
Código: HU4_CP2	Historia de usuario: 4
Nombre: Editar área	
Descripción: Prueba para la funcionalidad editar área.	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Administrador y además debe haber alguna área insertada en la base de datos.	
Entrada/ Pasos de ejecución: El usuario con rol Administrador introduce los datos requeridos para ser modificados en la base de datos. Para editar los datos debe ir al menú y seleccionar la opción "Área". En la ventana que se va a mostrar debe presionar la opción que indica editar un área. Una vez modificados los datos se presiona en el botón "Actualizar" y seguido se le muestra el área añadida.	
Resultado esperado: Mostrar un mensaje indicando el éxito de la operación.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 54: Caso de prueba de aceptación #8.

Caso de prueba de aceptación	
Código: HU4_CP3	Historia de usuario: 4
Nombre: Eliminar área	

Descripción: Prueba para la funcionalidad eliminar área.
Condiciones de ejecución: El usuario debe estar autenticado con el rol Administrador y además debe haber alguna área insertada en la base de datos.
Entrada/ Pasos de ejecución: Para eliminar un área debe ir al menú y seleccionar la opción "Área". En la ventana que se va a mostrar debe presionar la opción que indica eliminar un área, y se le pregunta al Oficial de Guardia Superior si está seguro que desea eliminar el área.
Resultado esperado: Mostrar un mensaje indicando el éxito de la operación.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 55: Caso de prueba de aceptación #9.

Caso de prueba de aceptación	
Código: HU4_CP4	Historia de usuario: 4
Nombre: Mostrar área	
Descripción: Prueba para la funcionalidad mostrar área.	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Administrador y además debe haber alguna área insertada en la base de datos.	
Entrada/ Pasos de ejecución: Para mostrar un área debe ir al menú y seleccionar la opción "Área". En la ventana que se va a mostrar debe presionar la opción que indica mostrar un área.	
Resultado esperado: Se mostrarán los datos del área seleccionada.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 56: Caso de prueba de aceptación #18.

Caso de prueba de aceptación	
Código: HU7_CP1	Historia de usuario: 7
Nombre: Crear hechos delictivos	
Descripción: Prueba para la funcionalidad crear hechos delictivos.	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Oficial de Guardia Superior para tener acceso a la funcionalidad y además debe haber alguna Guardia insertada en la base de datos.	
Entrada/ Pasos de ejecución: El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser insertado en la base de datos. Para insertar los datos debe ir al menú y seleccionar la opción "Guardia". En la ventana que se va a mostrar debe presionar en el botón "Hechos Delictivos". Una vez introducidos los datos se presiona en el botón "Crear" y seguido se le muestra el hecho delictivo añadido.	
Resultado esperado: Mostrar un mensaje indicando el éxito de la operación.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 57: Caso de prueba de aceptación #19.

Caso de prueba de aceptación	
Código: HU7_CP2	Historia de usuario: 7

Nombre: <i>Editar hechos delictivos</i>
Descripción: <i>Prueba para la funcionalidad editar hechos delictivos.</i>
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber algún hecho delictivo insertado en la base de datos.</i>
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser modificados en la base de datos. Para editar los datos debe ir al menú y seleccionar la opción “Guardia”, luego la opción “Hechos Delictivos” y por último la opción “Listado de Hechos Delictivos”. En la ventana que se va a mostrar debe presionar la opción que indica editar un hecho delictivo. Una vez modificados los datos se presiona en el botón “Actualizar” y seguido se le muestra el hecho delictivo modificado.</i>
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>
Evaluación de la prueba: <i>Prueba satisfactoria.</i>

Tabla 58: Caso de prueba de aceptación #20.

Caso de prueba de aceptación	
Código: <i>HU7_CP3</i>	Historia de usuario: <i>7</i>
Nombre: <i>Eliminar hechos delictivos</i>	
Descripción: <i>Prueba para la funcionalidad eliminar hechos delictivos.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber algún hecho delictivo insertado en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para eliminar un hecho delictivo debe ir al menú y seleccionar la opción “Guardia”, luego la opción “Hechos Delictivos” y por último la opción “Listado de Hechos Delictivos”. En la ventana que se va a mostrar debe presionar la opción que indica eliminar un hecho delictivo, y se le pregunta al Oficial de Guardia Superior si está seguro que desea eliminar el hecho delictivo.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 59: Caso de prueba de aceptación #21.

Caso de prueba de aceptación	
Código: <i>HU7_CP4</i>	Historia de usuario: <i>7</i>
Nombre: <i>Mostrar hechos delictivos</i>	
Descripción: <i>Prueba para la funcionalidad mostrar hechos delictivos.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber algún usuario insertado en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para eliminar un usuario debe ir al menú y seleccionar la opción “Guardia”, luego la opción “Hechos Delictivos” y por último la opción “Listado de Hechos Delictivos”. En la ventana que se va a mostrar debe presionar la opción que indica mostrar un hecho delictivo.</i>	
Resultado esperado: <i>Se mostrarán los datos del hecho delictivo seleccionado.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

3.5 Análisis de los resultados de la pruebas

En la metodología XP, la aplicación de pruebas se realiza de forma iterativa para comprobar la correcta implementación de las historias de usuario. Se realizaron un total de 41 pruebas funcionales para validar que el sistema funcionara correctamente, distribuidas en 3 iteraciones. A continuación se refleja en la Figura 12, la cantidad total de funcionalidades probadas y las no conformidades encontradas por cada iteración.

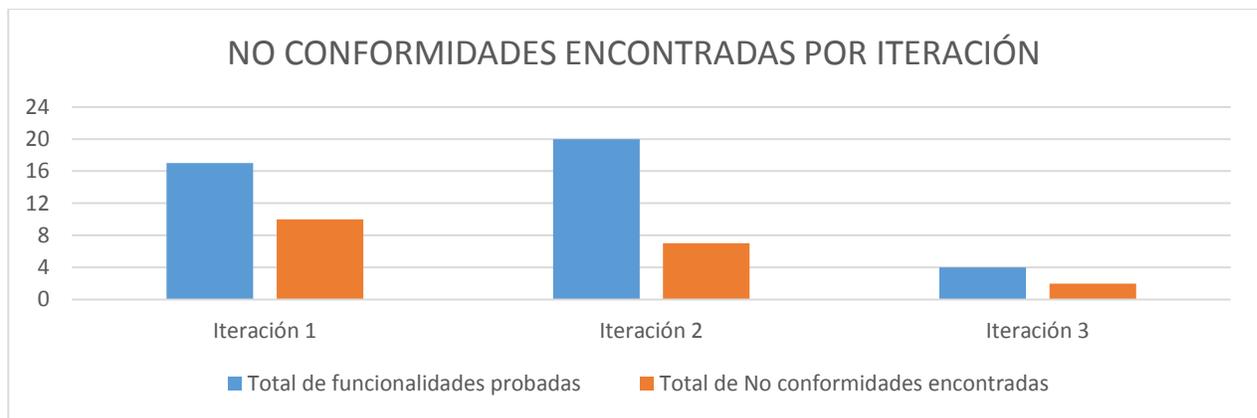


Figura 12: Total de funcionalidades probadas y no conformidades encontradas por iteración.

Finalmente todas las no conformidades fueron solucionadas, lo cual indica que se realizó un buen trabajo durante el proceso de prueba del sistema. Además se comprobó por el cliente la calidad de la propuesta de solución, logrando una mayor satisfacción por parte del cliente.

Conclusiones del capítulo

En el presente capítulo fueron definidos los estándares de codificación, los cuales proponen las mejores prácticas de programación, ayudando aún mejor entendimiento del código. Se describieron 56 tareas de ingeniería que ayudaron a implementar las HU. Se realizaron pruebas unitarias con la librería PHPUnit, demostrando el correcto funcionamiento de cada una de las funcionalidades de la aplicación. Además se realizaron las pruebas de aceptación para obtener un producto de mayor calidad, divididas en 3 iteraciones. En la primera iteración se analizaron un total de 17 pruebas, encontrando un total de 10 no conformidades referentes a validación del código, errores ortográficos y validación de entradas de datos incorrectos. En la segunda iteración se analizaron un total de 20 pruebas, encontrando un total de 7 no conformidades. Y por último en la tercera iteración se analizaron un total de 4 pruebas, encontrando un total de 2 no conformidades. Todas las no conformidades fueron corregidas por el equipo de desarrollo.

Conclusiones generales y Recomendaciones

Conclusiones generales

Con la realización de esta investigación se desarrolló un sistema informático capaz de mejorar el proceso de la gestión de la información de la Guardia Obrero Estudiantil en la Universidad de las Ciencias Informáticas. Los objetivos planteados fueron cumplidos de manera satisfactoria obteniendo a su vez una serie de resultados:

- El estudio del arte y el empleo de los métodos científicos posibilitaron identificar los conceptos y teorías que sustenta la investigación para el desarrollo de la solución.
- La selección de la metodología de desarrollo permitió al equipo de desarrollo cumplir con el objetivo general de la investigación y satisfacer de esta forma las necesidades del cliente.
- La selección de las herramientas, lenguajes de programación y tecnologías permitieron el desarrollo de la solución propuesta.
- La validación de la solución demostró el correcto funcionamiento del sistema.

Recomendaciones

Durante el desarrollo de la investigación y el desarrollo del sistema surgen aspectos que se deben tener en cuenta para el futuro perfeccionamiento del mismo, los cuales se sugieren a continuación:

- Desarrollar nuevas funcionalidades referentes a la representación gráfica, en aras de visualizar más la información referente al comportamiento de la Guardia Obrero Estudiantil.
- Incorporar nuevos módulos para realizar la planificación de la Guardia Obrero Estudiantil.

Referencias Bibliográficas

1. Sánchez Diez, Angel y Nocedo, Aismaray. Multimedia Interactiva Física General. Universidad de las Ciencias Informáticas. La Habana : s.n., 2008. Trabajo Diploma.
2. Gallardo Hidalgo, Yainelis y Milan Isaac, Miguel. Sistema de Información Geográfica para la Dirección de Seguridad y Protección en la Universidad de Ciencias Informáticas. La Habana : s.n., 2013. Trabajo de Diploma.
3. Promonegocios. [En línea] [Citado el: 23 de enero de 2015.] <http://www.promonegocios.net/mercadotecnia/que-es-informacion.html>.
4. FERRELL, O C y HIRT, GEOFFREY. Introducción a los Negocios en un Mundo Cambiante. Cuarta Edición. 2004.
5. MsC. Vidal Ledo, María Josefina y MsC. Araña Pérez, Ana Barbara. Gestión de la información y el conocimiento. [En línea] 2 de diciembre de 2012. [Citado el: 22 de enero de 2015.] <http://www.ems.sld.cu/index.php/ems/article/view/56/46>. Versión Electronica ISSN-1561-2902.
6. Ponjuán Dante, Gloria. Gestión de información en las organizaciones principios, conceptos y aplicaciones. Rosario : Ediciones Nuevo Paradigma, 2004.
7. Ingram, David. LaVozdeHouston. [En línea] [Citado el: 20 de enero de 2015.] http://pyme.lavoztx.com/Qué es un sistema de gestión de la información _ Pequeña y mediana empresa - La Voz Texas.htm.
8. Liz, Yimian de y Rivero, Soleydi. GestioPolis.com. [En línea] [Citado el: 10 de diciembre de 2014.] <http://www.gestiopolis.com/administracion-estrategia/sistemas-de-gestion-de-informacion-en-estudio-de-medio-ambiente.htm..>
9. Sistemas de Control de Rondas para Serenos. [En línea] [Citado el: 01 de Febrero de 2015.] <http://www.zuntini.com.ar/control%20de%20rondas.html>.
10. Sistema para el control de rondas para Guardias vigilantes. [En línea] [Citado el: 01 de Febrero de 2015.] <http://www.elipse.cl/productos/control%20de%20rondas/sistema%20control%20rondas%20Guardias%20vigilantes.html>.
11. Control de rondas de vigilancia mediante NFC. [En línea] [Citado el: 02 de Febrero de 2015.] <http://www.nfc-tracker.com/es/control-rondas-de-vigilancia/>.
12. Torres Aguilera, Grettell y Almeida Oquendo, Humberto. Gestor Web para el control de la Guardia Obrera de la Universidad de las Ciencias Informáticas (UCI). La Habana : s.n., 2008. Trabajo de Diploma.
13. Mata Sánchez, Dailín y Navarro Quintero, Aroldo. Sistema para la Gestión de la Guardia Obrera-Estudiantil. La Habana : s.n., 2013. Trabajo de Diploma.

14. Menendez Barzanallana, Rafael. Apuntes Infomática Aplicada a la Gestión Publica. Capítulo 2, Ingeniería del software. Metodologías de desarrollo. [En línea] 2011. [Citado el: 02 de Febrero de 2015.] <http://www.um.es/docencia/barzana/IAGP/IAGP2-Metodologias-de-desarrollo.html>.
15. Emunet.net. RUP Etapas Diseño. [En línea] [Citado el: 5 de Febrero de 2015.] <http://www.eumed.net/libros/2009c/584/RUP%20Diseno%20e%20implementacion%20del%20sistema.htm>.
16. Abrahamsson, P, y otros. New Directions on Agile Methods: A Comparative Analysis. 2003. IEEE.
17. Highsmith, James A. Agile Software Development Ecosystems. [En línea] 2002. [Citado el: 5 de Febrero de 2015.] http://www.google.com.cu/books?id=uE4FGFOHs2EC&pg=PP17&ots=OL0loD3o4q&dq=Agile%20software%20development%20ecosystems.%20&lr=lang_es%7Clang_en&hl=es&pg=PP17#v=onepage&q=Agile%20software%20development%20ecosystems.&f=false. ISBN/9780201760439.
18. Highsmith, J y Orr, K. Adaptive Software Development: A Collaborative Approach. s.l. : Dorset House, 2000.
19. Sutherland, Dr. Jeff. El Método SCRUM. [En línea] [Citado el: 10 de Febrero de 2015.] http://www.mastersoft.com.ar/MsWeb/otros_archivos/NotaScrumPCUsers.pdf..
20. Beck, K. Extreme Programming Explained. Embrace Change, Pearson Education. 1999.
21. Letelier, Patricio y Penandés, María Carmen. Metodologías ágiles para el desarrollo de software: eXtreme Programming(XP). [En línea] 15 de Abril de 2006. [Citado el: 5 de Febrero de 2015.] http://www.cyta.com.ar/ta0502/b_v5n2a1.htm.
22. Prf. Luza, César. Técnicas de modelamiento. Sesión 2: Introducción a UML. Lima, Perú : s.n., 2008.
23. Acosta Imbert, Odalis y Aimé, Rodríguez Martínez. SISTEMA DE GESTIÓN DE LA INFORMACIÓN PARA EL DIAGNÓSTICO TECNOLÓGICO A ENTIDADES LEGALES CUBANAS. La Habana : s.n., 2013.
24. Paradigm, Visual. Visual Paradigm Online Training Center. [En línea] 6 de Junio de 2012. [Citado el: 6 de Febrero de 2015.] http://www.visual-paradigm.com/support/documents/vpumluserguide/12/13/5963_aboutvisualp.html..
25. de programación, Lenguajes. El Mundo Informático. [En línea] 5 de Mayo de 2007. [Citado el: 6 de Febrero de 2015.] <http://jorgesaaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion>.
26. W3C. W3C Oficina Española. [En línea] [Citado el: 6 de Febrero de 2015.] <http://www.w3c.es/Divulgacion/a-z/#h>.
27. García Hernández, Alex y Roanny, Lamas López. Sistema para la planificación y control del entrenamiento en deportes de combate. La Habana : s.n., 2014.
28. W3C. W3C HojasEstilo. [En línea] [Citado el: 6 de Febrero de 2015.] <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>.

29. Luca, Damián De. HTML5:entienda el cambio, aproveche su potencial. [En línea] 2011. <http://books.google.com.cu/books?id=DGwjKIXWpm8C&lpg=PA29&dq=ventajas%20de%20css&hl=es&pg=PA30#v=onepage&q=ventajas%20de%20css&f=false>. ISBN 9789871773794.
30. Eguiluz Pérez, Javier. Introducción a JavaScript. 2009.
31. Pérez, L G. Lenguajes y Manejadores de Base de Datos.
32. Lafosse, Jerome. Struts 2 - El framework de desarrollo de aplicaciones. [En línea] [Citado el: 7 de Febrero de 2015.] http://books.google.com.cu/books?id=96HHRq6g5x8C&printsec=frontcover&dq=%C2%BFque+es+un+framework+de+desarrollo?&hl=es&sa=X&ei=ptqUYioF4GXtAbT_IG4BQ&ved=0CC0Q6AEwAA#.
33. Gutierrez, Javier J. Que es un framework web. [En línea] [Citado el: 7 de Febrero de 2015.] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
34. Amaury, Diogenes. JQuery. [En línea] 1 de Agosto de 2012. [Citado el: 10 de Febrero de 2015.] http://diogenesamaury.blogspot.com/2012_08_01_archive.html.
35. Builtwith. JavaScript Usage Statistics. Statistics for websites using JavaScript technologies. [En línea] 3 de Marzo de 2015. [Citado el: 3 de Marzo de 2015.] <http://trends.builtwith.com/javascript>.
36. Websites using JavaScript Frameworks. [En línea] [Citado el: 20 de Mayo de 2015.] <https://wappalyzer.com/categories/javascript-frameworks>.
37. VERMILION. Responsive CSS Framework Comparison. Bootstrap vs. Foundation vs. Skeleton. [En línea] 21 de Noviembre de 2013. [Citado el: 10 de Enero de 2015.] <http://www.scoop.it/t/responsive/p/3998820906/2013/03/24/responsive-css-framework-comparison-bootstrap-foundation-skeleton>.
38. Websites using Web Frameworks. [En línea] [Citado el: 20 de Mayo de 2015.] <https://wappalyzer.com/categories/web-frameworks>.
39. Torres, Michelle. Consultoría integral en tecnologías de la información . Qué es un framework y cuáles son los principales en PHP. [En línea] michelletorres.mx.
40. Eduardo Marreno, Michael, y otros. Extensión de la herramienta Visual Paradigm para la generación. . 2013. Vol. Volumen 6.
41. Groussard, Thierry. Recursos Informáticos Java Enterprise Edition - Desarrollo de aplicaciones web con JEE 6. s.l. : Ediciones ENI, 2010.
42. Welcome! - The Apache HTTP Server Project. [En línea] 2013. [Citado el: 7 de Febrero de 2015.] <http://httpd.apache.org/>.
43. Builtwith. Web Server Usage Statistics. Statistics for websites using Web Server technologies. [En línea] 3 de Marzo de 2015. <http://trends.builtwith.com/web-server>.
44. Wappalyzer. [En línea] 20 de Mayo de 2015. [Citado el: 20 de Mayo de 2015.] https://wappalyzer.com/categories/web-servers?utm_source=firefox&utm_medium=menu&utm_campaign=extensions..

45. [En línea] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd>.
46. Martínez, Rafael. Sobre PostgreSQL. [En línea] [Citado el: 7 de Febrero de 2015.] http://www.postgresql.org.es/sobre_postgresql.
47. NetBeans. NetBeans. [En línea] 2010. [Citado el: 10 de Enero de 2015.] http://netbeans.org/index_es.html.
48. Desarrollo de Software. [En línea] <http://www.download3k.es/Desarrollo-Software/Otros/Download-NetBeans-IDE.html>.
49. JEFFRIES, Ron, ANDERSON, Ann y HENDRICKSON, Chet. Extreme programming installed. s.l. : Addison-Esley professional, 2001. ISBN 0201708426.
50. Hillard, Rich. Recommended practice for architectural description of software-intensive system. s.l. : IEEE, 2000. págs. 16-20. Vol. 12.
51. Beck, Kent, Molina, Jesús García y Aguilar, Luis Loyanes. Una explicación de la programación extrema: Aceptar el cambio. s.l. : Addison Wesley, 2002. ISBN 8478290559.
52. Eguiluz, Javier. Introducción a Symfony 2. Desarrollo web ágil con Symfony 2. [En línea] 2013. [Citado el: 13 de Enero de 2015.]
53. Larman, Craig. UML y PATRONES. Introducción al análisis y diseño orientado a objetos. 2004. pág. 4.
54. Potencier, Fabien y Francois, Zaninoto. Symfony 1.4, la guía definitiva. Eguiluz, Javier. [En línea] [Citado el: 20 de Marzo de 2015.] http://librosweb.es/symfony_1_4.
55. Revisiones de código y estándares de codificación. [En línea] [Citado el: 16 de Abril de 2015.] <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
56. Gutierrez, Javier. Pruebas del Sistema en Programación Extrema. [En línea] [Citado el: 20 de Abril de 2015.] http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf.

Anexo 1: Historias de Usuario

Tabla 60: HU-Asignar rol.

Historia de usuario	
Número: 3	Nombre: Asignar rol
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 1 semana	Iteración Asignada: 1
Descripción: <i>El sistema permitirá que el Administrador pueda asignar un rol a un usuario mediante una interfaz interactiva. Los roles que los usuarios pueden desempeñar son los siguientes: Administrador, Oficial de Guardia Superior, Especialista de control y Directivo.</i>	
Observación: <i>El usuario tiene que estar autenticado con el rol Administrador.</i>	

Tabla 61: HU-Gestionar acercamiento de extranjeros.

Historia de usuario	
Número: 5	Nombre: Gestionar acercamiento de extranjeros
Usuario: Oficial de Guardia Superior	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 1 semana	Iteración Asignada: 1
Descripción: <i>El sistema permitirá que el Oficial de Guardia Superior pueda crear, editar, eliminar, mostrar y buscar un acercamiento de extranjero en la base de datos. Para esto se requiere:</i> <ul style="list-style-type: none"> <i>Descripción (Obligatorio)</i> 	
Observación: <i>El usuario tiene que estar autenticado con el rol Oficial de Guardia Superior.</i>	

Tabla 62: HU-Gestionar incidencias.

Historia de usuario	
Número: 6	Nombre: Gestionar incidencias
Usuario: Oficial de Guardia Superior	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 1 semana	Iteración Asignada: 1
Descripción: <i>El sistema permitirá que el Oficial de Guardia Superior pueda crear, editar, eliminar, mostrar y buscar una incidencia en la base de datos. Para esto se requiere:</i> <ul style="list-style-type: none"> <i>Descripción (Obligatorio)</i> 	
Observación: <i>El usuario tiene que estar autenticado con el rol Oficial de Guardia Superior.</i>	

Tabla 63: HU-Gestionar violaciones.

Historia de usuario

Número: 8	Nombre: Gestionar violaciones
Usuario: Oficial de Guardia Superior	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 1 semana	Iteración Asignada: 2
Descripción: El sistema permitirá que el Oficial de Guardia Superior pueda crear, editar, eliminar, mostrar y buscar eliminar una violación en la base de datos. Para esto se requiere: <ul style="list-style-type: none"> • Tipo (Obligatorio) • Descripción (Obligatorio) 	
Observación: El usuario tiene que estar autenticado con el rol Oficial de Guardia Superior.	

Tabla 64: HU-Gestionar afectaciones.

Historia de usuario	
Número: 9	Nombre: Gestionar afectaciones
Usuario: Oficial de Guardia Superior	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 1 semana	Iteración Asignada: 2
Descripción: El sistema permitirá que el Oficial de Guardia Superior pueda crear, editar, eliminar, mostrar y buscar eliminar una afectación en la base de datos. Para esto se requiere: <ul style="list-style-type: none"> • Tipo (Obligatorio) • Descripción (Obligatorio) 	
Observación: El usuario tiene que estar autenticado con el rol Oficial de Guardia Superior.	

Tabla 65: HU-Gestionar visitas.

Historia de usuario	
Número: 10	Nombre: Gestionar visitas
Usuario: Oficial de Guardia Superior	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 1 semana	Iteración Asignada: 2
Descripción: El sistema permitirá que el Oficial de Guardia Superior pueda crear, editar, eliminar, mostrar y buscar eliminar una visita en la base de datos. Para esto se requiere: <ul style="list-style-type: none"> • Tipo (Obligatorio) • Observación (Obligatorio) • Hora (Obligatorio) 	
Observación: El usuario tiene que estar autenticado con el rol Oficial de Guardia Superior.	

Tabla 66: HU-Gestionar extranjero.

Historia de usuario	
Número: 11	Nombre: Gestionar extranjero
Usuario: Oficial de Guardia Superior	

Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 1 semana	Iteración Asignada: 2
Descripción: El sistema permitirá que el Oficial de Guardia Superior pueda crear, editar, eliminar, mostrar y buscar eliminar un extranjero en la base de datos. Para esto se requiere: <ul style="list-style-type: none"> • Nombre (Obligatorio) • Apellidos (Obligatorio) • País (Obligatorio) 	
Observación: El usuario tiene que estar autenticado con el rol Oficial de Guardia Superior.	

Tabla 67: HU-Gestionar guardia.

Historia de usuario	
Número: 12	Nombre: Gestionar Guardia
Usuario: Oficial de Guardia Superior	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 1 semana	Iteración Asignada: 3
Descripción: El sistema permitirá que Oficial de Guardia Superior pueda crear, editar, eliminar, mostrar y buscar eliminar una Guardia en la base de datos. Para esto se requiere: <ul style="list-style-type: none"> • Fecha (Obligatorio) • Tipo (Obligatorio) • Turno (Obligatorio) • Presentes (Obligatorio) 	
Observación: El usuario tiene que estar autenticado con el rol Oficial de Guardia Superior.	

Tabla 68: HU-Generar reportes.

Historia de usuario	
Número: 14	Nombre: Generar reportes
Usuario: Especialista de control	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Punto de Estimación: 1.5 semana	Iteración Asignada: 3
Descripción: El sistema permitirá que el Especialista de control pueda generar un reporte mensual en la base de datos.	
Observación: El usuario tiene que estar autenticado con el rol Especialista de control.	

Tabla 69: HU-Consultar estadísticas.

Historia de usuario	
Número: 15	Nombre: Consultar estadísticas
Usuario: Directivo	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alto
Punto de Estimación: 1 semana	Iteración Asignada: 3

Descripción: <i>El sistema permitirá que los directivos puedan consultar las estadísticas.</i>
Observación: <i>El usuario tiene que estar autenticado con el rol Directivo.</i>

Tabla 70: HU-Exportar a PDF.

Historia de usuario	
Número: 16	Nombre: Exportar a PDF
Usuario: Directivo	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Alto
Punto de Estimación: 0.5 semana	Iteración Asignada: 3
Descripción: <i>El sistema permitirá que los directivos puedan exportar a PDF las estadísticas.</i>	
Observación: <i>El usuario tiene que estar autenticado con el rol Directivo.</i>	

Anexo 2: Tarjetas CRC

Tabla 71: Tarjeta CRC-Acercamiento de extranjeros.

Acercamiento de extranjeros	
Responsabilidad	Colaboración
Gestionar acercamiento de extranjeros	Guardia

Tabla 72: Tarjeta CRC-Incidencias.

Incidencias	
Responsabilidad	Colaboración
Gestionar incidencias	Guardia

Tabla 73: Tarjeta CRC-Violaciones.

Violaciones	
Responsabilidad	Colaboración
Gestionar violaciones	Guardia

Tabla 74: Tarjeta CRC-Afectaciones.

Afectaciones	
Responsabilidad	Colaboración
Gestionar afectaciones	Guardia

Tabla 75: Tarjeta CRC-Visitas.

Visitas

Responsabilidad	Colaboración
Gestionar visitas	Guardia

Tabla 76: Tarjeta CRC-Extranjero.

Extranjero	
Responsabilidad	Colaboración
Gestionar extranjero	Visitas

Tabla 77: Tarjeta CRC- Guardia.

Guardia	
Responsabilidad	Colaboración
Gestionar guardia	Parte diario
	Guardia

Anexo 3: Tareas de Ingeniería

Tabla 78: Tarea 7 de la HU Asignar rol.

Tarea de ingeniería	
Número: 7	Número de HU: 3
Nombre de la tarea: <i>Asignar rol</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.5
Fecha inicio: <i>8 de Febrero de 2015</i>	Fecha fin: <i>10 de Febrero de 2015</i>
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten asignar un rol a un usuario en la base de datos.</i>	

Tabla 79: Tarea 13 de la HU Gestionar Acercamiento de extranjeros.

Tarea de ingeniería	
Número: 13	Número de HU: 5
Nombre de la tarea: <i>Crear acercamiento de extranjeros</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: <i>17 de Febrero de 2015</i>	Fecha fin: <i>18 de Febrero de 2015</i>
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten crear un acercamiento de extranjeros en la base de datos.</i>	

Tabla 80: Tarea 14 de la HU Gestionar Acercamiento de extranjeros.

Tarea de ingeniería	
Número: 14	Número de HU: 5
Nombre de la tarea: <i>Editar acercamiento de extranjeros</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: <i>18 de Febrero de 2015</i>	Fecha fin: <i>19 de Febrero de 2015</i>
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten editar un acercamiento de extranjeros en la base de datos.</i>	

Tabla 81: Tarea 15 de la HU Gestionar Acercamiento de extranjeros.

Tarea de ingeniería	
Número: 15	Número de HU: 5
Nombre de la tarea: <i>Eliminar acercamiento de extranjeros</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: <i>20 de Febrero de 2015</i>	Fecha fin: <i>21 de Febrero de 2015</i>
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten eliminar un acercamiento de extranjeros en la base de datos.</i>	

Tabla 82: Tarea 16 de la HU Gestionar Acercamiento de extranjeros.

Tarea de ingeniería	
Número: 16	Número de HU: 5
Nombre de la tarea: <i>Mostrar acercamiento de extranjeros</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: <i>21 de Febrero de 2015</i>	Fecha fin: <i>22 de Febrero de 2015</i>
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten mostrar un acercamiento de extranjeros en la base de datos.</i>	

Tabla 83: Tarea 17 de la HU Gestionar Acercamiento de extranjeros.

Tarea de ingeniería	
Número: 17	Número de HU: 5
Nombre de la tarea: <i>Buscar acercamiento de extranjeros</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: <i>22 de Febrero de 2015</i>	Fecha fin: <i>23 de Febrero de 2015</i>
Programador responsable: <i>Raúl Rosabal Sánchez</i>	

Descripción: Se implementan las funcionalidades que permiten buscar un acercamiento de extranjeros en la base de datos.

Tabla 84: Tarea 18 de la HU Gestionar incidencias.

Tarea de ingeniería	
Número: 18	Número de HU: 6
Nombre de la tarea: <i>Crear incidencias</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 23 de Febrero de 2015	Fecha fin: 24 de Febrero de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten crear una incidencia en la base de datos.	

Tabla 85: Tarea 19 de la HU Gestionar incidencias.

Tarea de ingeniería	
Número: 19	Número de HU: 6
Nombre de la tarea: <i>Editar incidencias</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 24 de Febrero de 2015	Fecha fin: 25 de Febrero de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten editar una incidencia en la base de datos.	

Tabla 86: Tarea 20 de la HU Gestionar incidencias.

Tarea de ingeniería	
Número: 20	Número de HU: 6
Nombre de la tarea: <i>Eliminar incidencias</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 26 de Febrero de 2015	Fecha fin: 27 de Marzo de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten eliminar una incidencia en la base de datos.	

Tabla 87: Tarea 21 de la HU Gestionar incidencias.

Tarea de ingeniería	
Número: 21	Número de HU: 6
Nombre de la tarea: <i>Mostrar incidencias</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2

Fecha inicio: 27 de Febrero de 2015	Fecha fin: 28 de Febrero de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten mostrar una incidencia en la base de datos.	

Tabla 88: Tarea 22 de la HU Gestionar incidencias.

Tarea de ingeniería	
Número: 22	Número de HU: 6
Nombre de la tarea: <i>Buscar incidencias</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 28 de Febrero de 2015	Fecha fin: 1 de Marzo de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten buscar una incidencia en la base de datos.	

Tabla 89: Tarea 28 de la HU Gestionar violaciones.

Tarea de ingeniería	
Número: 28	Número de HU: 8
Nombre de la tarea: <i>Crear violaciones</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 7 de Marzo de 2015	Fecha fin: 8 de Marzo de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten crear una violación en la base de datos.	

Tabla 90: Tarea 29 de la HU Gestionar violaciones.

Tarea de ingeniería	
Número: 29	Número de HU: 8
Nombre de la tarea: <i>Editar violaciones</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 8 de Marzo de 2015	Fecha fin: 9 de Marzo de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten editar una violación en la base de datos.	

Tabla 91: Tarea 30 de la HU Gestionar violaciones.

Tarea de ingeniería	
Número: 30	Número de HU: 8
Nombre de la tarea: <i>Eliminar violaciones</i>	

Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 10 de Marzo de 2015	Fecha fin: 11 de Marzo de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten eliminar una violación en la base de datos.	

Tabla 92: Tarea 31 de la HU Gestionar violaciones.

Tarea de ingeniería	
Número: 31	Número de HU: 8
Nombre de la tarea: <i>Mostrar violaciones</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 11 de Marzo de 2015	Fecha fin: 12 de Marzo de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten mostrar una violación en la base de datos.	

Tabla 93: Tarea 32 de la HU Gestionar violaciones.

Tarea de ingeniería	
Número: 32	Número de HU: 8
Nombre de la tarea: <i>Buscar violaciones</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 12 de Marzo de 2015	Fecha fin: 13 de Marzo de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten buscar una violación en la base de datos.	

Tabla 94: Tarea 33 de la HU Gestionar afectaciones.

Tarea de ingeniería	
Número: 33	Número de HU: 9
Nombre de la tarea: <i>Crear afectaciones</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 14 de Marzo de 2015	Fecha fin: 15 de Marzo de 2015
Programador responsable: Raúl Rosabal Sánchez	
Descripción: Se implementan las funcionalidades que permiten crear una afectación en la base de datos.	

Tabla 95: Tarea 34 de la HU Gestionar afectaciones.

Tarea de ingeniería

Número: 34	Número de HU: 9
Nombre de la tarea: <i>Editar afectaciones</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 16 de Marzo de 2015	Fecha fin: 17 de Marzo de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten editar una afectación en la base de datos.</i>	

Tabla 96: Tarea 35 de la HU Gestionar afectaciones.

Tarea de ingeniería	
Número: 35	Número de HU: 9
Nombre de la tarea: <i>Eliminar afectaciones</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 17 de Marzo de 2015	Fecha fin: 18 de Marzo de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten eliminar una afectación en la base de datos.</i>	

Tabla 97: Tarea 36 de la HU Gestionar afectaciones.

Tarea de ingeniería	
Número: 36	Número de HU: 9
Nombre de la tarea: <i>Mostrar afectaciones</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 18 de Marzo de 2015	Fecha fin: 19 de Marzo de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten mostrar una afectación en la base de datos.</i>	

Tabla 98: Tarea 37 de la HU Gestionar afectaciones.

Tarea de ingeniería	
Número: 37	Número de HU: 9
Nombre de la tarea: <i>Buscar afectaciones</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 19 de Marzo de 2015	Fecha fin: 20 de Marzo de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten buscar una afectación en la base de datos.</i>	

Tabla 99: Tarea 38 de la HU Gestionar visitas.

Tarea de ingeniería	
Número: 38	Número de HU: 10
Nombre de la tarea: <i>Crear visitas</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 21 de Marzo de 2015	Fecha fin: 22 de Marzo de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten crear una visita en la base de datos.</i>	

Tabla 100: Tarea 39 de la HU Gestionar visitas.

Tarea de ingeniería	
Número: 39	Número de HU: 10
Nombre de la tarea: <i>Editar visitas</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 22 de Marzo de 2015	Fecha fin: 23 de Marzo de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten editar una visita en la base de datos.</i>	

Tabla 101: Tarea 40 de la HU Gestionar visitas.

Tarea de ingeniería	
Número: 40	Número de HU: 10
Nombre de la tarea: <i>Eliminar visitas</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 24 de Marzo de 2015	Fecha fin: 25 de Marzo de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten eliminar una visita en la base de datos.</i>	

Tabla 102: Tarea 41 de la HU Gestionar visitas.

Tarea de ingeniería	
Número: 41	Número de HU: 10
Nombre de la tarea: <i>Mostrar visitas</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 25 de Marzo de 2015	Fecha fin: 26 de Marzo de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten mostrar una visita en la base de datos.</i>	

Tabla 103: Tarea 42 de la HU Gestionar visitas.

Tarea de ingeniería	
Número: 42	Número de HU: 10
Nombre de la tarea: <i>Buscar visitas</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: <i>26 de Marzo de 2015</i>	Fecha fin: <i>27 de Marzo de 2015</i>
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten buscar una buscar en la base de datos.</i>	

Tabla 104: Tarea 43 de la HU Gestionar extranjero.

Tarea de ingeniería	
Número: 43	Número de HU: 11
Nombre de la tarea: <i>Crear extranjero</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.4
Fecha inicio: <i>28 de Marzo de 2015</i>	Fecha fin: <i>29 de Marzo de 2015</i>
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten crear un extranjero en la base de datos.</i>	

Tabla 105: Tarea 44 de la HU Gestionar extranjero.

Tarea de ingeniería	
Número: 44	Número de HU: 11
Nombre de la tarea: <i>Crear extranjero</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: <i>29 de Marzo de 2015</i>	Fecha fin: <i>30 de Marzo de 2015</i>
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten editar un extranjero en la base de datos.</i>	

Tabla 106: Tarea 45 de la HU Gestionar extranjero.

Tarea de ingeniería	
Número: 45	Número de HU: 11
Nombre de la tarea: <i>Eliminar extranjero</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: <i>31 de Marzo de 2015</i>	Fecha fin: <i>1 de Abril de 2015</i>
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten eliminar un extranjero en la base de datos.</i>	

Tabla 107: Tarea 46 de la HU Gestionar extranjero.

Tarea de ingeniería	
Número: 46	Número de HU: 11
Nombre de la tarea: <i>Mostrar extranjero</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 1 de Abril de 2015	Fecha fin: 2 de Abril de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten mostrar un extranjero en la base de datos.</i>	

Tabla 108: Tarea 47 de la HU Gestionar extranjero.

Tarea de ingeniería	
Número: 47	Número de HU: 11
Nombre de la tarea: <i>Buscar extranjero</i>	
Tipo de tarea: <i>Desarrollo</i>	Estimación: 0.2
Fecha inicio: 2 de Abril de 2015	Fecha fin: 3 de Abril de 2015
Programador responsable: <i>Raúl Rosabal Sánchez</i>	
Descripción: <i>Se implementan las funcionalidades que permiten buscar un extranjero en la base de datos.</i>	

Anexo 4: Pruebas de Aceptación

Tabla 109: Caso de prueba de aceptación #10.

Caso de prueba de aceptación	
Código: <i>HU5_CP1</i>	Historia de usuario: 5
Nombre: <i>Crear acercamiento de extranjeros</i>	
Descripción: <i>Prueba para la funcionalidad crear acercamiento de extranjeros.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior para tener acceso a la funcionalidad y además debe haber alguna Guardia insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser insertado en la base de datos. Para insertar los datos debe ir al menú y seleccionar la opción "Guardia". En la ventana que se va a mostrar debe presionar en el botón "Acercamiento de extranjeros". Una vez introducidos los datos se presiona en el botón "Crear" y seguido se le muestra el acercamiento de extranjeros añadido.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 110: Caso de prueba de aceptación #11.

Caso de prueba de aceptación	
Código: <i>HU5_CP2</i>	Historia de usuario: 5

Nombre: <i>Editar acercamiento de extranjeros</i>
Descripción: <i>Prueba para la funcionalidad editar acercamiento de extranjeros.</i>
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber algún acercamiento de extranjeros insertado en la base de datos.</i>
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser modificados en la base de datos. Para editar los datos debe ir al menú y seleccionar la opción “Guardia”, luego la opción “Acercamiento de extranjeros” y por último la opción “Listado de Acercamiento de extranjeros”. En la ventana que se va a mostrar debe presionar la opción que indica editar un acercamiento de extranjeros. Una vez modificados los datos se presiona en el botón “Actualizar” y seguido se le muestra el acercamiento de extranjeros modificado.</i>
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>
Evaluación de la prueba: <i>Prueba satisfactoria.</i>

Tabla 111: Caso de prueba de aceptación #12.

Caso de prueba de aceptación	
Código: <i>HU5_CP3</i>	Historia de usuario: <i>5</i>
Nombre: <i>Eliminar acercamiento de extranjeros</i>	
Descripción: <i>Prueba para la funcionalidad eliminar acercamiento de extranjeros.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber algún acercamiento de extranjeros insertado en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para eliminar un acercamiento de extranjeros debe ir al menú y seleccionar la opción “Guardia”, luego la opción “Acercamiento de extranjeros” y por último la opción “Listado de Acercamiento de extranjeros”. En la ventana que se va a mostrar debe presionar la opción que indica eliminar un acercamiento de extranjeros, y se le pregunta al Oficial de Guardia Superior si está seguro que desea eliminar el acercamiento de extranjeros.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 112: Caso de prueba de aceptación #13.

Caso de prueba de aceptación	
Código: <i>HU5_CP4</i>	Historia de usuario: <i>5</i>
Nombre: <i>Mostrar acercamiento de extranjeros</i>	
Descripción: <i>Prueba para la funcionalidad mostrar acercamiento de extranjeros.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber algún usuario insertado en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para mostrar un acercamiento de extranjeros debe ir al menú y seleccionar la opción “Guardia”, luego la opción “Acercamiento de extranjeros” y por último la opción “Listado de Acercamiento de extranjeros”. En la ventana que se va a mostrar debe presionar la opción que indica mostrar un acercamiento de extranjeros.</i>	
Resultado esperado: <i>Se mostrarán los datos del acercamiento de extranjeros seleccionado.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 113: Caso de prueba de aceptación #14.

Caso de prueba de aceptación	
Código: HU6_CP1	Historia de usuario: 6
Nombre: <i>Crear incidencias</i>	
Descripción: <i>Prueba para la funcionalidad crear incidencias y además debe haber alguna Guardia insertada en la base de datos.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior para tener acceso a la funcionalidad.</i>	
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser insertado en la base de datos. Para insertar los datos debe ir al menú y seleccionar la opción "Guardia". En la ventana que se va a mostrar debe presionar en el botón "Incidencias". Una vez introducidos los datos se presiona en el botón "Crear" y seguido se le muestra la incidencia añadida.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 114: Caso de prueba de aceptación #15.

Caso de prueba de aceptación	
Código: HU6_CP2	Historia de usuario: 6
Nombre: <i>Editar incidencias</i>	
Descripción: <i>Prueba para la funcionalidad editar incidencias.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna incidencia insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>El usuario con rol Administrador introduce los datos requeridos para ser modificados en la base de datos. Para editar los datos debe ir al menú y seleccionar la opción "Guardia", luego la opción "Incidencias y por último la opción "Listado de Incidencias". En la ventana que se va a mostrar debe presionar la opción que indica editar una incidencia. Una vez modificados los datos se presiona en el botón "Actualizar" y seguido se le muestra la incidencia modificada.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 115: Caso de prueba de aceptación #16.

Caso de prueba de aceptación	
Código: HU6_CP3	Historia de usuario: 6
Nombre: <i>Eliminar incidencias</i>	
Descripción: <i>Prueba para la funcionalidad eliminar incidencias.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna incidencia insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para eliminar una incidencia debe ir al menú y seleccionar la opción "Guardia", luego la opción "Incidencias y por último la opción "Listado de Incidencias". En la ventana</i>	

que se va a mostrar debe presionar la opción que indica eliminar una incidencia, y se le pregunta al Oficial de Guardia Superior si está seguro que desea eliminar la incidencia.

Resultado esperado: *Mostrar un mensaje indicando el éxito de la operación.*

Evaluación de la prueba: *Prueba satisfactoria.*

Tabla 116: Caso de prueba de aceptación #17.

Caso de prueba de aceptación	
Código: HU6_CP4	Historia de usuario: 6
Nombre: <i>Mostrar incidencias</i>	
Descripción: <i>Prueba para la funcionalidad mostrar incidencias.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna incidencia insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para mostrar una incidencia debe ir al menú y la opción "Guardia", luego la opción "Incidencias y por último la opción "Listado de Incidencias". En la ventana que se va a mostrar debe presionar la opción que indica mostrar una incidencia.</i>	
Resultado esperado: <i>Se mostrarán los datos de la incidencia seleccionada.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 117: Caso de prueba de aceptación #22.

Caso de prueba de aceptación	
Código: HU8_CP1	Historia de usuario: 8
Nombre: <i>Crear violaciones</i>	
Descripción: <i>Prueba para la funcionalidad crear violaciones.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior para tener acceso a la funcionalidad y además debe haber alguna Guardia insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser insertado en la base de datos. Para insertar los datos debe ir al menú y seleccionar la opción "Guardia". En la ventana que se va a mostrar debe presionar en el botón "Violaciones". Una vez introducidos los datos se presiona en el botón "Crear" y seguido se le muestra la violación añadida.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 118: Caso de prueba de aceptación #23.

Caso de prueba de aceptación	
Código: HU8_CP2	Historia de usuario: 8
Nombre: <i>Editar violaciones</i>	
Descripción: <i>Prueba para la funcionalidad editar violaciones.</i>	

Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna violación insertada en la base de datos.</i>
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser modificados en la base de datos. Para editar los datos debe ir al menú y seleccionar la opción "Guardia", luego la opción "Violaciones" y por último la opción "Listado de Violaciones". En la ventana que se va a mostrar debe presionar la opción que indica editar una violación. Una vez modificados los datos se presiona en el botón "Actualizar" y seguido se le muestra la violación modificada.</i>
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>
Evaluación de la prueba: <i>Prueba satisfactoria.</i>

Tabla 119: Caso de prueba de aceptación #24.

Caso de prueba de aceptación	
Código: <i>HU8_CP3</i>	Historia de usuario: <i>8</i>
Nombre: <i>Eliminar violaciones</i>	
Descripción: <i>Prueba para la funcionalidad eliminar violaciones.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna violación insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para eliminar una violación debe ir al menú y seleccionar la opción "Guardia", luego la opción "Violaciones" y por último la opción "Listado de Violaciones". En la ventana que se va a mostrar debe presionar la opción que indica eliminar una violación, y se le pregunta al Oficial de Guardia Superior si está seguro que desea eliminar la violación.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 120: Caso de prueba de aceptación #25.

Caso de prueba de aceptación	
Código: <i>HU8_CP4</i>	Historia de usuario: <i>8</i>
Nombre: <i>Mostrar violaciones</i>	
Descripción: <i>Prueba para la funcionalidad mostrar violaciones.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna violación insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para mostrar una violación debe ir al menú y seleccionar la opción "Guardia", luego la opción "Violaciones" por último la opción "Listado de Violaciones". En la ventana que se va a mostrar debe presionar la opción que indica mostrar una violación.</i>	
Resultado esperado: <i>Se mostrarán los datos de la violación seleccionada.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 121: Caso de prueba de aceptación #26.

Caso de prueba de aceptación

Código: HU9_CP1	Historia de usuario: 9
Nombre: <i>Crear afectaciones</i>	
Descripción: <i>Prueba para la funcionalidad crear afectaciones.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior para tener acceso a la funcionalidad y además debe haber alguna Guardia insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser insertado en la base de datos. Para insertar los datos debe ir al menú y seleccionar la opción "Guardia". En la ventana que se va a mostrar debe presionar en el botón "Afectaciones". Una vez introducidos los datos se presiona en el botón "Crear" y seguido se le muestra la afectación añadida.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 122: Caso de prueba de aceptación #27.

Caso de prueba de aceptación	
Código: HU9_CP2	Historia de usuario: 9
Nombre: <i>Editar afectaciones</i>	
Descripción: <i>Prueba para la funcionalidad editar afectaciones.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna afectación insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser modificados en la base de datos. Para editar los datos debe ir al menú y seleccionar la opción "Guardia", luego la opción "Afectaciones" y por último la opción "Listado de Afectaciones". En la ventana que se va a mostrar debe presionar la opción que indica editar una afectación. Una vez modificados los datos se presiona en el botón "Actualizar" y seguido se le muestra la afectación modificada.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 123: Caso de prueba de aceptación #28.

Caso de prueba de aceptación	
Código: HU9_CP3	Historia de usuario: 9
Nombre: <i>Eliminar afectaciones</i>	
Descripción: <i>Prueba para la funcionalidad eliminar afectaciones.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna afectación insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para eliminar un usuario debe ir al menú y seleccionar la opción "Guardia", luego la opción "Afectaciones" y por último la opción "Listado de Afectaciones". En la ventana que se va a mostrar debe presionar la opción que indica eliminar una afectación, y se le pregunta al Oficial de Guardia Superior si está seguro que desea eliminar la afectación.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	

Evaluación de la prueba: *Prueba satisfactoria.*

Tabla 124: Caso de prueba de aceptación #29.

Caso de prueba de aceptación	
Código: HU9_CP4	Historia de usuario: 9
Nombre: <i>Mostrar afectaciones</i>	
Descripción: <i>Prueba para la funcionalidad mostrar afectaciones.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna afectación insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para mostrar una afectación debe ir al menú y seleccionar la opción "Guardia", luego la opción "Afectaciones" y por último la opción "Listado de Afectaciones". En la ventana que se va a mostrar debe presionar la opción que indica mostrar una afectación.</i>	
Resultado esperado: <i>Se mostrarán los datos de la afectación seleccionada.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 125: Caso de prueba de aceptación #30.

Caso de prueba de aceptación	
Código: HU10_CP1	Historia de usuario: 10
Nombre: <i>Crear visitas</i>	
Descripción: <i>Prueba para la funcionalidad crear visitas.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior para tener acceso a la funcionalidad y además debe haber alguna Guardia insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser insertado en la base de datos. Para insertar los datos debe ir al menú y seleccionar la opción "Guardia". En la ventana que se va a mostrar debe presionar en el botón "Visitas". Una vez introducidos los datos se presiona en el botón "Crear" y seguido se le muestra la visita añadida.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 126: Caso de prueba de aceptación #31.

Caso de prueba de aceptación	
Código: HU10_CP2	Historia de usuario: 10
Nombre: <i>Editar visitas</i>	
Descripción: <i>Prueba para la funcionalidad editar visitas.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna visita insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser modificados en la base de datos. Para editar los datos debe ir al menú y seleccionar la opción "Guardia", luego la opción "Visitas" y por último la opción "Listado de Visitas". En la ventana</i>	

que se va a mostrar debe presionar la opción que indica editar una visita. Una vez modificados los datos se presiona en el botón “Actualizar” y seguido se le muestra la visita modificada.

Resultado esperado: *Mostrar un mensaje indicando el éxito de la operación.*

Evaluación de la prueba: *Prueba satisfactoria.*

Tabla 127: Caso de prueba de aceptación #32.

Caso de prueba de aceptación	
Código: HU10_CP3	Historia de usuario: 10
Nombre: <i>Eliminar visitas</i>	
Descripción: <i>Prueba para la funcionalidad eliminar visitas.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna visita insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para eliminar una visita debe ir al menú y seleccionar la opción “Guardia”, luego la opción “Visitas” y por último la opción “Listado de Visitas”. En la ventana que se va a mostrar debe presionar la opción que indica eliminar una visita, y se le pregunta al Oficial de Guardia Superior si está seguro que desea eliminar la visita.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 128: Caso de prueba de aceptación #33.

Caso de prueba de aceptación	
Código: HU10_CP4	Historia de usuario: 10
Nombre: <i>Mostrar visitas</i>	
Descripción: <i>Prueba para la funcionalidad mostrar visitas.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna visita insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para mostrar una visita debe ir al menú y seleccionar la opción “Guardia”, luego la opción “Visitas” y por último la opción “Listado de Visitas”. En la ventana que se va a mostrar debe presionar la opción que indica mostrar una visita.</i>	
Resultado esperado: <i>Se mostrarán los datos de la visita seleccionada.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 129: Caso de prueba de aceptación #34.

Caso de prueba de aceptación	
Código: HU11_CP1	Historia de usuario: 11
Nombre: <i>Crear extranjeros</i>	
Descripción: <i>Prueba para la funcionalidad crear extranjeros.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Administrador para tener acceso a la funcionalidad y además debe haber alguna visita insertada en la base de datos.</i>	

Entrada/ Pasos de ejecución: <i>El usuario con rol Administrador introduce los datos requeridos para ser insertado en la base de datos. Para insertar los datos debe seleccionar la opción “Visitas” y luego la opción “Listado de Visitas”. En la ventana que se va a mostrar debe presionar en el botón “Extranjeros”. Una vez introducidos los datos se presiona en el botón “Crear” y seguido se le muestra el extranjero añadido.</i>
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>
Evaluación de la prueba: <i>Prueba satisfactoria.</i>

Tabla 130: Caso de prueba de aceptación #35.

Caso de prueba de aceptación	
Código: HU11_CP2	Historia de usuario: 11
Nombre: <i>Editar extranjeros</i>	
Descripción: <i>Prueba para la funcionalidad editar extranjeros.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber algún extranjero insertado en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser modificados en la base de datos. Para editar los datos seleccionar la opción “Visitas”, a continuación la opción “Listado de Visitas”, luego la opción “Hechos Delictivos” y por último la opción “Listado de Extranjeros”. En la ventana que se va a mostrar debe presionar la opción que indica editar un extranjero. Una vez modificados los datos se presiona en el botón “Actualizar” y seguido se le muestra el extranjero modificado.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 131: Caso de prueba de aceptación #36.

Caso de prueba de aceptación	
Código: HU11_CP3	Historia de usuario: 11
Nombre: <i>Eliminar extranjeros</i>	
Descripción: <i>Prueba para la funcionalidad eliminar extranjeros.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber algún extranjero insertado en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para eliminar un extranjero debe seleccionar la opción “Visitas”, a continuación la opción “Listado de Visitas”, luego la opción “Hechos Delictivos” y por último la opción “Listado de Extranjeros”. En la ventana que se va a mostrar debe presionar la opción que indica eliminar un extranjero, y se le pregunta al Oficial de Guardia Superior si está seguro que desea eliminar el extranjero.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 132: Caso de prueba de aceptación #37.

Caso de prueba de aceptación

Código: HU11_CP4	Historia de usuario: 11
Nombre: <i>Mostrar extranjeros</i>	
Descripción: <i>Prueba para la funcionalidad mostrar extranjeros.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber algún extranjero insertado en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>Para eliminar un usuario debe seleccionar la opción "Visitas", a continuación la opción "Listado de Visitas", luego la opción "Hechos Delictivos" y por último la opción "Listado de Extranjeros". En la ventana que se va a mostrar debe presionar la opción que indica mostrar un extranjero.</i>	
Resultado esperado: <i>Se mostrarán los datos del extranjero seleccionado.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 133: Caso de prueba de aceptación #38.

Caso de prueba de aceptación	
Código: HU12_CP1	Historia de usuario: 12
Nombre: <i>Crear guardia</i>	
Descripción: <i>Prueba para la funcionalidad crear guardia.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior para tener acceso a la funcionalidad.</i>	
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser insertado en la base de datos. Para insertar los datos debe ir al menú y seleccionar la opción "Guardia". En la ventana que se va a mostrar debe presionar en el botón "Nuevo". Una vez introducidos los datos se presiona en el botón "Crear" y seguido se le muestra la guardia añadida.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 134: Caso de prueba de aceptación #39.

Caso de prueba de aceptación	
Código: HU12_CP2	Historia de usuario: 12
Nombre: <i>Editar guardia</i>	
Descripción: <i>Prueba para la funcionalidad editar guardia.</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna guardia insertada en la base de datos.</i>	
Entrada/ Pasos de ejecución: <i>El usuario con rol Oficial de Guardia Superior introduce los datos requeridos para ser modificados en la base de datos. Para editar los datos debe ir al menú y seleccionar la opción "Guardia". En la ventana que se va a mostrar debe presionar la opción que indica editar una guardia. Una vez modificados los datos se presiona en el botón "Actualizar" y seguido se le muestra la guardia modificada.</i>	
Resultado esperado: <i>Mostrar un mensaje indicando el éxito de la operación.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

Tabla 135: Caso de prueba de aceptación #40.

Caso de prueba de aceptación	
Código: HU12_CP3	Historia de usuario: 12
Nombre: Eliminar guardia	
Descripción: Prueba para la funcionalidad eliminar guardia.	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna guardia insertada en la base de datos.	
Entrada/ Pasos de ejecución: Para eliminar una guardia debe ir al menú y seleccionar la opción "Guardia". En la ventana que se va a mostrar debe presionar la opción que indica eliminar una guardia, y se le pregunta al Oficial de Guardia Superior si está seguro que desea eliminar la guardia.	
Resultado esperado: Mostrar un mensaje indicando el éxito de la operación.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 136: Caso de prueba de aceptación #41.

Caso de prueba de aceptación	
Código: HU12_CP4	Historia de usuario: 12
Nombre: Mostrar guardia	
Descripción: Prueba para la funcionalidad mostrar guardia.	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Oficial de Guardia Superior y además debe haber alguna guardia insertada en la base de datos.	
Entrada/ Pasos de ejecución: Para mostrar una guardia debe ir al menú y seleccionar la opción "Guardia". En la ventana que se va a mostrar debe presionar la opción que indica mostrar una guardia.	
Resultado esperado: Se mostrarán los datos de la guardia seleccionada.	
Evaluación de la prueba: Prueba satisfactoria.	