

Universidad de Holguín
“Oscar Lucero Moya”
Facultad de Informática y Matemática



Trabajo para optar por el título de Ingeniero en Informática

*Sistema para la gestión de información de la guardia obrera y
estudiantil en la Universidad de Holguín “Oscar Lucero Moya”*

Autora: Tahimi Ronda Almaguer

Tutora: MsC. Jenny Ruiz de la Peña

Holguín, Cuba
Julio, 2010

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo a la Casa de Software de la Facultad de Informática y Matemática de la Universidad de Holguín para que hagan el uso que estimen pertinente con este trabajo. Para que así conste firmo la presente a los 9 días del mes de julio de 2010.

Firma de la Autora

Firma de la Tutora

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado **Sistema para la gestión de información de la guardia obrera y estudiantil en la Universidad de Holguín "Oscar Lucero Moya"** fue realizado en la Facultad de Informática y Matemática de la Universidad de Holguín. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

Totalmente

Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____.

Pensamiento

*A veces vale más la tenacidad que el genio,
la perseverancia que la capacidad, porque en tanto que el genio y la inteligencia
pueden vacilar en la lucha, cansarse y abandonar el terreno,
la tenacidad prosigue poco a poco, pero con seguridad,
hasta que llega a la meta deseada.*

Braulio Pérez Marcio

Agradecimientos

A Dios, por ser el que ha guiado mi camino y ha hecho que esté hoy aquí.

A mis padres, por ser lo más bello que tengo, por su creencia constante de que podía lograrlo.

A mis tíos Roxana y Juan, por sus rezos para que llegara este momento.

A Alberto, por su apoyo constante y por darme fuerzas para seguir adelante en todo momento.

A mi tutora, por su ayuda incondicional, por hacer de este trabajo la obra excelente que es hoy.

A todos mis compañeros de aula, en especial a Yolanda por su gran ayuda, a Dianela, María Isabel y Ariagna por su constante ayuda y por darme fuerzas.

A todos los que me han brindado su ayuda incondicional y constantemente han estado preocupados por la prosperidad de este trabajo.

A mis profesores.

Dedicatoria

A mis padres.

A mi esposo.

A mis suegros.

A mi tutora.

A mis compañeros de curso.

*A todas las personas que de una forma u otra me han dado su mano
incondicionalmente.*

Resumen

En la Universidad de Holguín "Oscar Lucero Moya" se planifica, organiza y se realiza la guardia obrera y estudiantil, como parte de la necesidad de incorporar a sus trabajadores y estudiantes al cumplimiento del deber de cuidar la propiedad pública y social. El servicio de guardia, tiene como objetivo fundamental contribuir a la seguridad y protección de los bienes, recursos y personas. Actualmente, el proceso de gestión de información de los servicios de guardia realizados por estudiantes y trabajadores resulta complejo y difícil de realizar. La información se almacena de forma manual y en formato Word, por lo que es complicado hacer búsquedas y actualizaciones. La forma en que se conserva, no garantiza protección y no asegura su disponibilidad para un acceso futuro.

En el presente trabajo se describe la investigación realizada en el proceso de gestión de información de la guardia obrera y estudiantil, con el objetivo de mejorar este, y lograr que sea de una manera más eficiente. Con el desarrollo del Sistema para la gestión de información de la guardia obrera y estudiantil, se logró un medio que unificara los datos y garantizara la integridad y consistencia de los mismos. Es posible acceder a la información fácilmente, y gestionarla de forma óptima.

Para la elaboración de la aplicación Web se utilizó el lenguaje de programación Java con el apoyo del framework para aplicaciones Web enriquecidas ZK, como servidor Web el Apache Tomcat y PostgreSQL como Sistema Gestor de Base de Datos. El sistema obtenido es multiplataforma.

Abstract

At “Oscar Lucero Moya” University in Holguín, the students and employees’ guard duty is planned, organized and done as part of the necessity to involve employees and students in the fulfillment of this duty concerned with the protection of social and public property. The main goal of this service is to contribute to the safeguard and protection of property, resources and people. Currently, the information management process concerned with the guard duty done by employees and students turns complex and difficult to do. Information is gathered manually and on a Word format what makes search and updates complicated. The way the information is saved, does not guarantee its protection or ensures its protection for its availability if future access is required.

In this work the research done in the process of information management of employees and students’ guard duty is described with the purpose to improve it and thus make it more efficient. With the development of the System for the Management of employees and students’ guard duty, a means to unify all data and to guarantee their interconnection and consistency. It makes possible to access to the information and to manage it in a highly efficient way.

For the elaboration of the Web application, Java programming language was used as supported on ZK enriched Web applications, Apache Tomcat as Web server, and PostgreSQL as Data Base Managing System. The system obtained is a multipath one.

Índice

Introducción	1
Capítulo 1 Fundamentación Teórica	9
1.1 Descripción general de la Guardia Obrera y Estudiantil.	9
1.1.1 Características de la GOE en la UHo.....	9
1.1.2 Procesos del negocio	10
1.1.3 Análisis de la ejecución de los procesos del negocio.....	14
1.2 Sistemas informáticos existentes vinculados al campo de acción	14
1.3 Tendencias y tecnologías actuales.....	16
1.3.1 Arquitectura cliente / servidor.....	16
1.3.2 Aplicaciones Web	17
1.3.3 Software libre.....	18
1.3.4 Lenguajes de programación Web	19
1.3.5 Framework.....	24
1.3.6 Patrón Modelo Vista Controlador.....	26
1.3.7 Tecnologías de Servidores Web.....	26
1.3.8 Sistemas Gestores de Bases de Datos	27
1.3.9 Metodologías para el desarrollo de software	30
1.3.9.1 Proceso Unificado de Rational	30
1.3.9.2 Metodologías ágiles	32
1.4 Análisis crítico de las fuentes y bibliografía utilizada	33
1.5 Conclusiones del capítulo	34
Capítulo 2 Descripción de la solución propuesta	35
2.1 Modelación del Negocio	35
2.2 Reglas del negocio.....	35
2.3 Descripción de los procesos del negocio	36
2.3.1 Actores y trabajadores del negocio.....	37
2.3.2 Diagrama de Casos de Uso del Negocio	38
2.3.3 Diagramas de actividades de los casos de uso del negocio.....	39
2.3.4 Diagramas de clases del Modelo de Objeto	41
2.4 Modelación del sistema.....	41
2.4.1 Diagrama de paquetes del sistema	41

2.4.2 Actores del sistema	43
2.4.3 Requerimientos del sistema.....	44
2.4.3.1 Requerimientos funcionales	44
2.4.3.2 Requerimientos no funcionales	47
2.4.4 Diagrama de casos de uso del sistema	49
2.5 Estudio de sostenibilidad.....	50
2.5.1 Factibilidad administrativa, técnica, ambiental y sociocultural	51
2.5.1.1 Cálculo del esfuerzo y tiempo de desarrollo.....	55
2.5.1.2 Cálculo del costo.....	57
2.6 Base de Datos.....	61
2.6.1 Diagrama de Clases Persistentes y Modelo de Datos.....	61
2.6.2 Gestor de Base de datos	63
2.7 Diagrama de clases del diseño	63
2.8 Diseño Visual	65
2.9 Diagrama de despliegue	66
2.10 Diagrama de componentes	67
2.11 Seguridad.....	69
2.12 Integración con otras aplicaciones vinculadas a la gestión de información.....	69
2.13 Aplicación del criterio de expertos.....	70
2.14 Conclusiones del capítulo	73
Conclusiones Generales.....	74
Recomendaciones	75
Bibliografía.....	76
Glosario de Términos	80
Anexos.....	I

Índice de Tablas

Tabla 2.1 Actores del negocio	37
Tabla 2.2 Trabajadores del negocio	38
Tabla 2.3 Actores del sistema.....	43
Tabla 2.4 Entradas externas (EI)	52
Tabla 2.5 Salidas externas (EO)	53
Tabla 2.6 Peticiones (EQ).....	54
Tabla 2.7 Ficheros internos (ILF).....	54
Tabla 2.8 Puntos de función desajustados	55
Tabla 2.9 Características	55
Tabla 2.10 Multiplicadores de esfuerzo	56
Tabla 2.11 Factores de escala.....	56
Tabla 2.12 Aspectos considerados muy relevantes por los expertos consultados.	71
Tabla 2.13 Aspectos considerados muy relevantes por los usuarios consultados.	72

Índice de Figuras

Figura 2.1 Diagrama de Casos de Uso del Negocio	39
Figura 2.2 Diagrama de actividad CUN Confeccionar Plan de Guardia Obrera y Estudiantil .	40
Figura 2.3 Diagrama de paquetes del sistema	42
Figura 2.4 Actores del sistema	44
Figura 2.5 Diagrama de casos de uso del sistema	50
Figura 2.6 Diagrama de clases persistentes.....	62
Figura 2.7 Diagrama de clases del diseño CUS Actualizar Guardia Estudiantil	65
Figura 2.8 Pantalla de bienvenida de la aplicación.....	66
Figura 2.9 Diagrama de despliegue de la aplicación	67
Figura 2.10 Diagrama de componentes Paquete Indicadores	68

Introducción

Las Tecnologías de la Informática y las Comunicaciones (TIC) han revolucionado significativamente la existencia del hombre y su desarrollo político, social y económico, convirtiéndose en la actualidad en un pilar fundamental en la realización de toda actividad humana. Su uso aporta numerosas ventajas, que vienen dadas de acuerdo con sus características fundamentales, como son: la inmaterialidad, la interactividad, la instantaneidad, la innovación, elevados parámetros de calidad de imagen y sonido, digitalización, influencia, más sobre los procesos que sobre los productos, automatización, interconexión y diversidad [43].

Muchas son las empresas e instituciones que trazan estrategias para lograr informatizar sus labores. En este proceso, las universidades cubanas, han planteado el desarrollo de sistemas informáticos que faciliten la gestión de información, así como la informatización de procesos para lograr un mejor funcionamiento y rendimiento de estas instituciones de acuerdo con las necesidades actuales.

En las últimas décadas, se ha generalizado la utilización de ambos medios: la computación y las comunicaciones, para lograr un progresivo avance en la gestión de la información y la posibilidad de administrarla desde cualquier lugar. De ahí la existencia de sitios y aplicaciones Web como expresión de un sistema de información, que pretende facilitar la organización y el acceso a los contenidos y servicios disponibles. Esas tecnologías se presentan cada vez más como una necesidad en el contexto de sociedad donde los rápidos cambios, el aumento de los conocimientos y las demandas de una educación de alto nivel constantemente actualizada se convierten en una exigencia permanente. Por su parte las universidades del país, factor clave dentro de la educación, no están exentos de estos avances [51].

La Universidad de Holguín, con 35 años de fundada y dedicada a la formación de futuros profesionales de la sociedad cubana, es uno de los Centros de Educación Superior (CES) de mayor matrícula y protagonismo en la provincia de Holguín. Esta entidad cuenta actualmente con dos sedes universitarias. La Sede Universitaria "Oscar Lucero Moya" consta de 4 facultades las cuales son: Facultad de Ingeniería, Facultad de Ciencias Económicas,

Facultad de Informática y Matemática, y la Facultad de Ingeniería Industrial; y la Sede Universitaria "Celia Sánchez Manduley ", está compuesta por las facultades de Ciencias Sociales, Humanidades y Derecho.

Entre las tareas sustantivas que realiza este CES se encuentran la actividad de pregrado para las carreras vinculadas a las facultades y el curso regular por encuentro para la superación de los trabajadores. Se realiza el trabajo docente en las Centros Universitarios Municipales y se atiende la actividad de ciencia y técnica. Se destaca la formación de doctores y másteres así como proyectos de investigación con líneas muy bien definidas. Por otra parte se tiene en cuenta la extensión universitaria. Esto se vincula con la Residencia Estudiantil con un conjunto de actividades de relevancia nacional, lo cual va en marcha ascendente. Además, se cuenta con un área de administración y servicios que incluye el hotel Turacade, casa de visita para la atención de los visitantes.

La Universidad de Holguín está estructurada, además de las áreas de dirección y administración, por departamentos docentes y no docentes en cada una de las facultades, centros de estudio y laboratorios correspondientes a cada una de las carreras, y las áreas independientes. En esta entidad, los estudiantes y trabajadores, tienen como compromiso ineludible, el deber de cuidar y preservar las instalaciones y bienes que les han sido asignados para realizar sus actividades. Es por eso que resulta necesario garantizar debidamente la seguridad y protección física en este Centro de Educación Superior.

En todos los centros educacionales del país se presta profunda atención a la organización, realización puntual y efectiva de la guardia obrera y la vigilancia revolucionaria, que junto a los grupos de enfrentamiento al delito, hechos y manifestaciones de corrupción e ilegalidades, velan por preservar los recursos del pueblo y no permitir hacer el juego a los enemigos de la Revolución.

Es deber insustituible e histórico de la clase obrera y de sus trabajadores custodiar y proteger los bienes y recursos del Estado, de los cuales deben ser sus principales veladores. Sin desconocer la necesidad de la protección profesional en determinados sectores de la

economía, la guardia obrera y la vigilancia revolucionaria han constituido y constituyen una de las principales armas en la defensa de la Revolución.

En la Universidad de Holguín, se planifica, organiza y se realiza la guardia obrera y estudiantil, como parte de la necesidad de incorporar a sus trabajadores y estudiantes al cumplimiento del deber de cuidar la propiedad pública y social. Corresponde a los cuadros responsables establecer los objetivos a cuidar, horarios, lugares donde se realiza la guardia, y crear las condiciones necesarias para que se ejecute.

El Plan de Guardia Obrera y Estudiantil constituye el documento rector para la ejecución de esta trascendental tarea, en el cual se plasma toda la distribución de la guardia obrera en el año. Primeramente se solicitan a las áreas y departamentos el potencial obrero, que es la relación de los trabajadores que los integran. A partir de los datos que se entregan se realiza la asignación desde septiembre hasta julio del año próximo.

El Plan de Guardia Obrera y Estudiantil, una vez confeccionado, se envía directamente a los Jefes de Áreas, Jefes de Departamentos y Decanos de las facultades, para que se dé a conocer la distribución realizada. Estas personas lo revisan, y si detectan dificultades o errores informan estos señalamientos para que los mismos sean corregidos. Los Jefes de Áreas y Departamentos son los encargados de informar a los trabajadores los horarios y frecuencias de guardia correspondientes a cada persona y garantizar que se presenten a la realización de la misma.

En el caso de la guardia estudiantil, a cada facultad le es asignada una cantidad determinada de días que se mantiene constante todos los meses del curso, a partir de la matrícula de los estudiantes. La Federación Estudiantil Universitaria (FEU) junto a los coordinadores de año y profesores guías, son los responsables de distribuir los estudiantes, informar la relación de los que están de guardia y velar por el cumplimiento de la misma.

Es importante tener organizada la relación de los potenciales que envían las áreas, y departamentos para la confección del plan de guardia, y efectuar de manera segura y ordenada la elaboración del mismo, de forma tal que no existan incoherencias a la hora de

distribuir las personas con sus horarios y frecuencias. Además, es necesario poder hacer modificaciones sin provocar alteraciones en el resto de la distribución, posibilitar la obtención de reportes por Jefes de Departamentos y Áreas, y verificar el cumplimiento de la guardia por parte de trabajadores y estudiantes.

Actualmente, el registro y control de toda esta información se realiza en formato Word y a veces manualmente. En ocasiones no fluye con la rapidez y calidad requerida e incluso, el control que se tiene de ella no es el adecuado. La información es almacenada con poca seguridad lo que provoca que los datos no sean totalmente confiables, puedan estar incorrectos y se repita la misma información. Se generan grandes volúmenes de documentación, pues es necesario preservar los datos para la obtención de informes de planes de guardias anteriores. Esto trae consigo en gran medida que toda esta información no esté organizada adecuadamente y sea en ocasiones imprecisa. Resulta complejo y a la vez lento hacer búsquedas y realizar actualizaciones, además de obtener datos específicos. El modo en que se conserva no ofrece garantía de protección, no asegura posibilidades de acceso futuro y dificulta la actualización y corrección de los datos almacenados.

A partir de la problemática existente se deriva el **problema de investigación**: ¿Cómo mejorar la gestión de información de la guardia obrera y estudiantil en la Universidad de Holguín “Oscar Lucero Moya”?

Por tanto el **objeto de estudio** se corresponde con el proceso de gestión de información de la guardia obrera y estudiantil en la Universidad de Holguín “Oscar Lucero Moya”.

Se propone como **objetivo de investigación**: desarrollar una aplicación Web, para mejorar la gestión de información relacionada con la guardia obrera y estudiantil en la Universidad de Holguín “Oscar Lucero Moya”.

El objetivo delimita el **campo de acción**: informatización del proceso de gestión de información de la guardia obrera y estudiantil en la Universidad de Holguín “Oscar Lucero Moya”.

Para guiar esta investigación se plantea la siguiente **hipótesis**: una aplicación Web, que gestione eficientemente el flujo de información, que brinde a sus datos seguridad y confiabilidad garantizando su preservación, actualización, corrección y adecuada organización, caracterizado por la rapidez en los procesos, favorecerá la gestión de información referente a la guardia obrera y estudiantil.

El desarrollo de la investigación transcurrió a través de las siguientes **tareas**:

1. Estudio de los procesos que tienen lugar en la universidad respecto al control y planificación de la guardia obrera y estudiantil.
2. Evaluación de la situación existente en la gestión de información de la guardia obrera y estudiantil.
3. Estudio de las principales tendencias y tecnologías para el desarrollo de aplicaciones Web.
4. Captura de requerimientos del producto informático.
5. Valoración de sostenibilidad del producto informático según las dimensiones administrativa, socio-humanista, ambiental y tecnológica.
6. Diseño e implementación del producto informático.
7. Validación del producto informático mediante el criterio de expertos.

Para realizar la investigación se utilizaron los siguientes **métodos científicos**:

Métodos Teóricos:

Análisis y Síntesis: se utilizó para descomponer el problema existente en pequeños subproblemas a fin de un mayor entendimiento, y la síntesis para integrarlos en el desarrollo del sistema que dará solución a la problemática.

Histórico Lógico: se utilizó para conocer la historia y evolución de los procesos relacionados con la guardia obrera y estudiantil, y el funcionamiento y desarrollo de los mismos.

Hipotético Deductivo: se tomó como premisa la hipótesis planteada para a partir de ella llegar a conclusiones sobre los requerimientos necesarios que debía tener el sistema para solucionar el problema y cumplir las necesidades de los usuarios.

Modelación: muy útil para el diseño y desarrollo de la aplicación Web, se aplicó para representar cada una de las fases del diseño del sistema informático, pues se empleó la metodología de ingeniería de software Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés), la cuál emplea el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés).

Enfoque sistémico: proporcionó una orientación general para el estudio de los procesos presentes en la situación problemática, analizándose todo este conjunto de procesos como un sistema general, integrado por componentes, los cuales cumplen determinadas funciones y mantienen una interacción entre ellos.

Métodos Empíricos:

La entrevista: constituyó una técnica de recopilación de información con la que se pudo adquirir información acerca de lo que se investigaba. Permitted definir claramente la situación problemática, se usó para obtener información en la investigación en general y en varias fases del proceso de diseño.

La observación: se utilizó en la etapa inicial para el diagnóstico del problema a investigar, lo que fue de gran utilidad en el diseño de la investigación. Permitted conocer cómo se desarrollaban los procesos y el flujo de información, así como el funcionamiento del proceso de confección del Plan de Guardia Obrera y Estudiantil.

Revisión de documentos: se utilizó para la recopilación de información, el estudio de diferentes bibliografías para el diseño y desarrollo del sistema, además para definir los requerimientos funcionales y analizar la organización de la información del plan de guardia.

Encuesta: se aplicó para conocer y evaluar el grado de satisfacción de los usuarios finales, mediante el criterio de expertos, lo que permitió demostrar la validez o no de la hipótesis planteada.

Para el diseño del sistema se aplicaron todas las etapas correspondientes a la metodología Proceso Unificado de Rational (RUP, por sus siglas en inglés), empleando la Herramienta para Asistir por Computadora a la Ingeniería de Software (CASE, por sus siglas en inglés) Rational Rose.

Para su desarrollo, se utilizó el lenguaje de programación Java, con el apoyo del Framework ZK, herramienta para facilitar y agilizar el desarrollo de aplicaciones Web. Para almacenar la información, se optó por el Sistema Gestor de Base de Datos PostgreSQL y como servidor Web el Apache Tomcat. El sistema desarrollado es multiplataforma.

Se realizó una búsqueda a fin de encontrar documentación sobre el tema de la guardia obrera y estudiantil. No obstante no fueron encontradas muchas bibliografías que trataran el mismo. En general se encontró información en algunos periódicos nacionales que trataban aspectos acerca de la importancia y necesidad de la realización de esta importante tarea.

La mayor parte de la información fue obtenida en las entrevistas realizadas al personal encargado de todo este proceso en la universidad. Además se consultaron algunos reglamentos para la guardia obrera y estudiantil, a partir de los cuales se obtuvo información sobre aspectos esenciales en la ejecución y cumplimiento de la misma.

Se encontró un sistema informático para el control de la guardia obrera y estudiantil, desarrollado en la Facultad de Construcciones de la Universidad de Oriente. Este sistema permite la planificación de la guardia obrera y estudiantil de forma automática, el acceso a historiales de los trabajadores y el cálculo del por ciento de asistencia a la misma. No obstante, a pesar de las ventajas que reporta por sus características, presenta ciertas limitaciones en sus funcionalidades respecto a la forma de organización de la guardia, por lo que no puede ser aplicado a la universidad. En general no se encontraron suficientes

antecedentes en sistemas informáticos para la gestión de información de la guardia obrera y estudiantil con las especificaciones necesarias.

El presente documento está estructurado por introducción, dos capítulos, conclusiones, recomendaciones, bibliografía y anexos.

El capítulo 1 “**Fundamentos Teóricos**” describe los principales procesos relacionados con la guardia obrera y estudiantil, expone las principales tendencias y tecnologías para el desarrollo de aplicaciones Web haciendo énfasis en las utilizadas para la creación de la solución, así como una crítica de la bibliografía consultada.

En el capítulo 2 “**Descripción de la solución propuesta**” se plantean los requerimientos funcionales y no funcionales, se realiza la valoración de sostenibilidad del producto informático según las dimensiones administrativa, socio-humanista, ambiental y tecnológica, y el diseño de la base de datos y de la aplicación Web en general.

Por último, se presentan las Conclusiones Generales a las que se arribaron, las Recomendaciones que se proponen, la Bibliografía utilizada, y los Anexos con información adicional sobre el presente trabajo.

Capítulo 1 Fundamentación Teórica

En el presente capítulo se exponen los fundamentos teóricos que sustentan la presente investigación. En primer lugar se describen las características y los principales procesos relacionados con la guardia obrera y estudiantil en la Universidad de Holguín, así como un análisis de la ejecución de los mismos. Se explican las principales tendencias y tecnologías para el desarrollo de aplicaciones Web, haciendo énfasis en las seleccionadas para la construcción de la solución propuesta. Además se hace una breve descripción de la metodología de desarrollo a utilizar y se realiza una crítica de la bibliografía consultada.

1.1 Descripción general de la Guardia Obrera y Estudiantil.

Existen entidades estatales cuyo objeto social o aprobado son los servicios de seguridad y protección. Estas actividades también son realizadas por grupos de Seguridad Interna, los serenos y los trabajadores y estudiantes durante la realización de la guardia obrera y estudiantil. Los organismos, órganos y entidades, en coordinación con las organizaciones sindicales y estudiantiles consideran como parte del Sistema de Seguridad y Protección Física, la ejecución de la guardia obrera y estudiantil como complemento para fortalecer la vigilancia, en aquellos lugares donde esta se organice [36].

Los estudiantes y trabajadores de la Educación Superior, tienen como compromiso insoslayable con la Patria, el deber de cuidar y preservar las instalaciones y bienes que les han sido asignados a los centros en que realizan sus actividades. Es por eso que resulta necesario garantizar debidamente la seguridad y protección física en los Centros de Educación Superior.

1.1.1 Características de la GOE en la UHo

En la Universidad de Holguín "Oscar Lucero Moya" se planifica, organiza y se realiza la guardia obrera y estudiantil, como parte de la necesidad de incorporar a sus trabajadores y estudiantes al cumplimiento del deber de cuidar la propiedad pública y social. Corresponde a los cuadros responsables establecer los objetivos a cuidar, horarios, lugares donde se realiza la guardia, y crear las condiciones necesarias para que se ejecute [48].

Los trabajadores y estudiantes de la Universidad "Oscar Lucero Moya" tienen como compromiso ineludible el cuidar y proteger los bienes y recursos que el Estado ha puesto en sus manos para el desempeño social de la institución, y resulta necesario, por tanto, garantizar debidamente la Seguridad y Protección en el centro y sus dependencias. El servicio de guardia obrera y estudiantil se realiza por los trabajadores y estudiantes del centro y tiene como objetivo fundamental contribuir a la seguridad y protección de los bienes, recursos y personas. La prestación de este servicio requiere, del personal que lo realiza, una alta preparación y el estricto cumplimiento de todas las disposiciones que se establecen por la entidad.

Los trabajadores que laboran en el centro, por decisión expresa y voluntaria, ratifican su compromiso de participar en el cumplimiento del servicio de guardia, ajustada a un estricto cumplimiento de todas las tareas y misiones que les son asignadas. Los estudiantes matriculados en el curso regular diurno, con excepción de los extranjeros, tienen el deber de cumplir con la guardia, para lo que realizan su compromiso a través de las facultades y la FEU.

1.1.2 Procesos del negocio

La Universidad de Holguín "Oscar Lucero Moya" lleva a cabo la realización puntual y efectiva de la guardia obrera y estudiantil, siendo responsable de la organización y control de esta actividad el Jefe de Seguridad y Protección de la entidad, el cual tiene a su cargo esta importante tarea, así como la preparación e información de los trabajadores. La ejecución de la guardia se realiza a partir de la organización y distribución del personal, y se lleva un registro estricto de toda la distribución y funciones a cumplir. Los principales procesos relacionados con la planificación y el control de la guardia obrera y estudiantil que se ejecutan en la universidad son: la solicitud del potencial obrero, la elaboración del Plan de Guardia Obrera y Estudiantil, la organización en las facultades de la guardia estudiantil y el registro del cumplimiento de la guardia obrera y estudiantil.

La guardia obrera y estudiantil, es organizada y controlada a partir de una estructura definida por el Jefe de Seguridad y Protección de la entidad, que es la persona que se encuentra al frente de esta elemental tarea. La distribución de las principales funciones, y la relación de

todo el personal de las áreas y facultades con los horarios y frecuencias establecidos, están ubicadas en el Plan de Guardia Obrera y Estudiantil, documento que rige este importante compromiso.

Todos los cursos, en el mes de junio, en las reuniones de trabajo realizadas en la universidad, se solicitan a Vicerrectores, Decanos, Jefes de departamentos y Jefes de Áreas, el listado del potencial obrero para la planificación del siguiente curso escolar. Se entrega un modelo de potencial a llenar con los siguientes datos: número de orden, nombre y apellidos, sexo, departamento o área, sugerencias de Ayudante de oficial de Guardia o Patrulla Fin de Semana, y observaciones. Este modelo es entregado al departamento de Seguridad y Protección para la confección del Plan de Guardia Obrera.

El departamento de Seguridad y Protección clasifica la relación de los potenciales según el tipo de guardia a prestar. La guardia obrera se organiza en los siguientes tipos de servicio:

- Oficiales de Guardia Superiores
- Ayudantes del Oficial de Guardia Operativa
- Guardia Obrera de Fin de Semana

Como Oficiales de Guardia Superiores se designan a los principales cuadros y dirigentes de la dirección de la universidad. Realizan el control y dirección de la guardia obrera y estudiantil, con un día fijo mensualmente durante todo el curso. Los datos almacenados en el Plan de Guardia son: nombre y apellidos, cargo, fecha y facultad o área. Los días de guardia de estas personas, están en correspondencia con los de la facultad a la que pertenecen, para la realización de la guardia estudiantil, a fin de lograr una mayor comunicación entre profesores y estudiantes.

Los Ayudantes del Oficial de Guardia Operativa prestan el servicio de guardia nocturna los días de semana, desde las 07:00 p.m. hasta las 03:00 a.m., con la misión fundamental de realizar el control y organización de la guardia estudiantil y el cumplimiento del resto de las tareas que indique el Oficial de Guardia Operativa. Este servicio de guardia se presta por profesores y trabajadores no docentes con la capacidad requerida para el desempeño de estas funciones en horario nocturno (personal masculino).

La Guardia Obrera de Fin de Semana se realiza los sábados y domingos en 2 turnos, de 08:00 a.m. a 12:00 p.m. y de 12:00 p.m. hasta las 04:00 p.m. Se realiza fundamentalmente por el personal femenino y algunos profesores y trabajadores con ciertas limitaciones para la guardia nocturna. La Guardia Estudiantil es planificada por las facultades, en base a la cantidad de estudiantes y carreras, designándose 13 estudiantes diarios para su prestación, organizados en 2 turnos: 6 estudiantes de 07:00 p.m. a 11:00 p.m. y 7 estudiantes de 11:00 p.m. a 02:00 a.m. Para esto el departamento de Seguridad y Protección designa a cada facultad una fecha que es fija mensualmente para prestar el servicio. En cada facultad la guardia estudiantil es responsabilidad de la FEU (Federación Estudiantil Universitaria) y los decanos. A su vez es organizada por el (la) Jefe(a) de Carrera junto a los coordinadores de año y profesores guías, los que distribuyen los estudiantes según los grupos, informan los correspondientes horarios y deben garantizar el cumplimiento de la misma. La relación de los estudiantes que están de guardia debe presentarse al puesto de mando 24 horas antes de la realización de la misma.

La universidad cuenta con un puesto de dirección (Puesto de Mando) que lleva todo el control de la realización de la guardia obrera y estudiantil. En este local se realiza diariamente el servicio de guardia por un técnico de la defensa y defensa civil, el Oficial de Guardia Operativa. Se lleva un registro diario del personal de guardia en un modelo establecido que recoge: área, fecha, número de orden, nombre y apellidos, posta, hora de entrada, hora de salida, evaluación y firma.

Diariamente se lleva a cabo la recepción y entrega de la guardia con la participación de la dotación entrante y saliente, bajo la dirección del jefe del departamento de Seguridad y Protección. Esta dotación está integrada por el Oficial de Guardia Operativa, Jefe de Grupo Seguridad Interna, Jefe de Turno del Grupo Seguridad Interna y el Profesor Educativo Docente de la Residencia Estudiantil. Este puesto de dirección, además de las tareas de control y dirección del servicio de guardia obrera-estudiantil, responde por la recepción, recopilación e información a la Rectora y al Puesto de Mando del Ministerio de Educación Superior. Igualmente responde por las tareas de la defensa que se desarrollan, como una de las principales la recepción y desencadenamiento del aviso en los distintos grados de

disposición para la defensa, así como para las distintas situaciones (alarma aérea sobre situación radioactiva o química y el ataque terrestre) y también el aviso a la brigada de respuesta rápida ante alteraciones del orden o disturbios contrarrevolucionarios.

Semanalmente se realiza el control del cumplimiento de la guardia obrera y estudiantil en las reuniones de trabajo y en los consejos de dirección, siendo responsabilidad del jefe del departamento de Seguridad y Protección brindar esa información; este resultado del comportamiento de la guardia se informa del 1 al 5 de cada mes al Ministerio de Educación Superior y a los rectores de las universidades del país.

El Plan de Guardia Obrera y Estudiantil se elabora para todo el curso escolar, a partir de los potenciales de guardia, y recoge todos los datos de los trabajadores, los distintos cargos específicos y generales y la relación de horarios y frecuencias de los diferentes tipos de guardia. Primeramente se seleccionan los Oficiales Superiores, los trabajadores de guardia nocturna y luego se distribuye la Guardia Obrera de Fin de Semana. Cada trabajador y estudiante realiza aproximadamente de 2 a 4 guardias en todo el curso. Una vez confeccionado, se envía a las facultades, departamentos y áreas. Los decanos y jefes de áreas lo revisan, y en caso de que existan errores, bajas o nuevas situaciones con determinados datos o trabajadores, mandan al departamento de Seguridad y Protección estos señalamientos, los cuales son arreglados y estos cambios son informados nuevamente al personal que está involucrado en los mismos. Toda esta información circula a través del correo electrónico.

Los jefes de departamento y jefes de áreas se encargan de informar a cada profesor los días y horarios de guardia, y son responsables de que todo el personal esté debidamente enterado con un tiempo de antelación. En caso de cambios de turnos de guardia entre profesores esto se hace de manera extraoficial, internamente en el departamento. El puesto de mando de la universidad, lleva un control diario del personal que está de guardia, a partir de un ejemplar del plan de guardia destinado al local para el cumplimiento de sus funciones. Se informa en cada turno al trabajador o estudiante la posta correspondiente, y luego se procede a registrar el cumplimiento o no de la guardia. La misión fundamental es velar por un

correcto funcionamiento de la guardia obrera y estudiantil, la asistencia a la misma y el cuidado y preservación de los bienes y recursos de la universidad.

1.1.3 Análisis de la ejecución de los procesos del negocio

Actualmente, el proceso de gestión de información de la guardia obrera y estudiantil, no se realiza de la manera más eficiente, y existen inconvenientes en el flujo y manipulación de la información. La forma en que se conserva, no garantiza protección, no asegura su disponibilidad para un acceso futuro, y hace más compleja la actualización y corrección de los datos almacenados.

Las relaciones de los potenciales obreros se entregan en hojas de papel, lo que provoca que se acumule una gran cantidad de datos, y a la vez es necesario teclear los mismos para confeccionar el Plan de Guardia Obrera y Estudiantil. La elaboración del mismo se realiza en forma de tablas en documentos Word, lo que hace difícil organizar la información, y muchas veces existen incoherencias a la hora de distribuir las personas con sus horarios y frecuencias. Además, todo esto provoca una gran lentitud a la hora de hacer búsquedas y actualizaciones. Toda esta información circula a través del correo electrónico, por lo que ante fallos del servidor o la red, puede suceder que no llegue a tiempo o a determinadas personas.

El control del cumplimiento de la guardia se lleva a cabo de forma manual, mediante un modelo impreso que es llenado diariamente, esto hace que se acumule una gran cantidad de información en papeles, por lo que existe el riesgo de que estos se deterioren y consecuentemente se produzca pérdida de información. Además resulta molesto llevar el registro de los datos y la vez tener un correcto dominio de toda la información.

1.2 Sistemas informáticos existentes vinculados al campo de acción

La evolución de la informática en aras de lograr adelantos en la sociedad actual, ha sido marcada fuertemente por el desarrollo acelerado de nuevas tecnologías de la información y las comunicaciones. En este aspecto, el desarrollo de sistemas informáticos se ha convertido en una herramienta fundamental para las instituciones, con el objetivo principal de lograr una mayor eficiencia en los procesos y garantizar mejores resultados.

Teniendo en cuenta lo anterior, se realizó una investigación acerca de los sistemas informáticos para la gestión de información de la guardia obrera y estudiantil que se utilizan en Cuba y el mundo. A partir de la información obtenida se valoró que no existen suficientes antecedentes de este tipo de sistemas actualmente.

No se encontraron sitios Web que de alguna forma divulgaran o brindaran algún tipo de datos respecto a este tema. Algunos periódicos nacionales contenían artículos que daban información sobre las misiones y características de la guardia obrera. Solo se encontró un sistema informático para el control de la guardia obrera y estudiantil, desarrollado en la Facultad de Construcciones de la Universidad de Oriente [54]. El mismo contempla las siguientes funcionalidades:

- Planificación de la guardia obrera y estudiantil.
- Acceso al historial de cada trabajador y estudiante.
- Registro del cumplimiento de la guardia
- Cálculo del por ciento de asistencia

Para el desarrollo de este sistema se empleó el lenguaje de programación Procesador de Hipertexto (PHP, por sus siglas en inglés) y MySQL como Sistema de Administración de Bases de Datos.

No obstante a sus funcionalidades, el sistema presenta algunos inconvenientes. Las interfaces con que fue diseñado no tienen un acceso directo al resto de las funcionalidades de una forma explícita, además presenta ciertas limitaciones para el control de los datos. El sistema solo contempla la guardia estudiantil de los estudiantes externos, y clasifica la guardia de los trabajadores de manera distinta a la que se realiza en la universidad, los horarios y frecuencias son diferentes y están predefinidos, por lo que no pueden modificarse. Contiene los departamentos y áreas de esa facultad específicamente, por lo que las estadísticas de cumplimiento solo se remiten a estos recintos. En general la planificación de la guardia se organiza de manera distinta a la universidad y rigurosamente adaptada a sus características particulares, por lo que este sistema no es posible aplicarlo en la Universidad de Holguín.

1.3 Tendencias y tecnologías actuales

1.3.1 Arquitectura cliente / servidor

En las tecnologías actuales a partir del desarrollo de la computación, están presentes varias arquitecturas para la utilización de software y hardware, entre estas, la arquitectura cliente/servidor. El término cliente/servidor es originalmente aplicado a la arquitectura de software que describe el procesamiento entre dos o más programas: una aplicación y un servicio soportante. La arquitectura de este tipo es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema [57].

Características del modelo cliente / servidor

1. Las funciones de cliente y servidor pueden estar en plataformas separadas, o en la misma plataforma.
2. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los clientes o de los servidores se realizan de una manera transparente para el usuario final.
3. La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
4. Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones clientes.
5. Muchos clientes pueden utilizar un mismo servicio sin tenerlo, lo que ahorra espacio y disminuye la carga de trabajo del cliente.

Arquitectura multi-capas

La arquitectura cliente/servidor genérica tiene dos tipos de nodos en la red: clientes y servidores. Consecuentemente, estas arquitecturas genéricas se refieren a veces como arquitecturas de dos niveles o dos capas. La red puede disponer de tres tipos de nodos, los clientes que interactúan con los usuarios, servidores de aplicación que procesan los datos y servidores de la base de datos que almacenan los datos para los servidores de aplicación; en este caso se trata de una arquitectura de tres-capas.

Ventajas de la arquitectura cliente / servidor

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio [57].

1.3.2 Aplicaciones Web

La arquitectura cliente/servidor es utilizada en numerosos servicios de cómputo, para lograr un mejor procesamiento y rendimiento, además de garantizar más eficacia en el uso de herramientas para el tratamiento de datos, entre las que se encuentran las aplicaciones Web. Una aplicación Web es aquella que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador. Es una aplicación software que se codifica en un lenguaje soportado por los navegadores Web (Lenguaje de Marcas Hipertextuales o HTML por sus siglas en inglés, JavaScript, Java, etc.), en la que se confía la ejecución al navegador [58]. Es una de las herramientas más eficaces para visualizar y gestionar información y se logra una mayor eficiencia en el tratamiento y manejo de la misma.

Ventajas y Beneficios de las aplicaciones Web

El uso de las aplicaciones Web proporciona numerosas ventajas a la hora de almacenar y manipular información, ya que permiten de una manera más eficiente la obtención y actualización de datos, además de la seguridad y confiabilidad en el acceso a los mismos. Facilitan el trabajo a distancia, pues solo es necesaria una computadora con un navegador Web que cumpla los requerimientos necesarios y es posible trabajar desde cualquier computador portátil con conexión a Internet, además no se requieren complicadas combinaciones de hardware/software para ser utilizadas.

Las aplicaciones Web proveen una alta disponibilidad, ya que se pueden realizar consultas en cualquier parte del mundo donde se tenga acceso a Internet y a cualquier hora. No es necesario instalar ni desarrollar una aplicación en el cliente, cualquier cambio solo se tiene que actualizar en el servidor. Permiten tener un mayor control de los datos y más seguridad en las diferentes secciones de las mismas, un avanzado sistema de consultas, altas, bajas y modificaciones de datos, lo cual mantiene la información siempre actualizada. Otorgan la flexibilidad de determinar niveles de acceso según la confidencialidad de la información guardada.

1.3.3 Software libre

En la actualidad, a nivel mundial crecen cada vez más las expectativas por la utilización de software libre a partir de los beneficios que trae consigo. Son numerosas las personas y organizaciones que trazan estrategias para la migración hacia el mismo. En Cuba, el uso de la Informática ha estado soportado en sistemas Windows; actualmente Linux, ha ganado espacios como sistema libre y de código abierto lo cual le confiere numerosas ventajas. Por esta razón el país se prepara para realizar la migración hacia este [28].

El software libre, en general, se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, proporciona a los usuarios del software cuatro libertades: el uso del programa con cualquier propósito; el estudio de su funcionamiento, y su adaptación a las necesidades; la distribución de copias, con lo cual se puede ayudar a otros y mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie [34].

El uso de software libre denota numerosas ventajas para los usuarios. Es posible corregir errores de manera rápida y dinámica, ya que muchas personas tienen acceso al código fuente. Se puede continuar ofreciendo soporte, desarrollo u otro tipo de servicios para el software. Ante errores ortográficos y gramaticales, estos pueden ser subsanados con mayor rapidez por una persona capacitada. Las personas pueden ver y entender cómo se almacenan los datos en un determinado formato o sistema. Se puede usar aun después de que haya desaparecido la persona que lo elaboró, dado que cualquier técnico informático puede continuar desarrollándolo, mejorándolo o adaptándolo. Se logra un ahorro en costos de adquisición ya que se puede ejercer con la compra de una sola licencia, implantación y hay una disminución significativa en el costo de soporte, sin embargo esto no ocurre con los costos de mantenimiento y de interoperabilidad.

Significación del uso del Software Libre para Cuba

Sin duda alguna, el uso del Software Libre es sustentable en Cuba, a partir de las ventajas que tiene con respecto a los del tipo propietario. Por esto, su aplicación como plataforma informática de trabajo adquiere una relevante significación que puede verse desde tres ámbitos diferentes:

Político: Representa la no utilización de productos informáticos que demanden la autorización de sus propietarios (licencias) para su explotación.

Económico: Su utilización no implica gastos adicionales por concepto de cambio de plataforma de software, por cuanto es operable en el mismo soporte de hardware con que cuenta el país.

Tecnológico: Permite su adaptación a los contextos de aplicación, al contar con su código fuente, lo cual garantiza un mayor por ciento de efectividad, además la corrección de sus errores de programación y obtención de las actualizaciones y nuevas versiones.

1.3.4 Lenguajes de programación Web

Actualmente, el software libre es ampliamente utilizado a partir de los beneficios que reporta, ya que es posible utilizar tecnologías avanzadas y eficientes para el desarrollo de sistemas informáticos de acuerdo a las necesidades actuales como los lenguajes de programación Web. En el caso de las aplicaciones Web, estas generan dinámicamente una serie de

páginas en un formato estándar, como HTML o Lenguaje Extensible de Marcado de Hipertexto (XHTML, por sus siglas en inglés), soportadas por los navegadores Web comunes. Se utilizan lenguajes interpretados en el lado y del cliente, para añadir elementos dinámicos a la interfaz de usuario. Durante la sesión, el navegador Web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación Web. Ejemplos: JavaScript, Visual Basic Script, Lenguaje Marcado de Hipertexto Dinámico (DHTML, por sus siglas en inglés) [56].

También se utilizan lenguajes de lado servidor, que son aquellos lenguajes que se caracterizan por implementar la lógica del negocio dentro del servidor, además de encargarse de la tarea del acceso a las bases de datos y el tratamiento de la información. Ejemplos: Páginas Activas del Servidor (ASP, por sus siglas en inglés), Procesador de Hipertexto (PHP, por sus siglas en inglés) y Páginas de Servidor de Java (JSP, por sus siglas en inglés). A continuación se relacionan algunas características de algunos de estos lenguajes utilizados para la creación de aplicaciones Web, haciendo énfasis en el lenguaje de programación Java y ZK Framework, utilizado en desarrollo Web bajo plataforma Java, herramientas por las cuales se optó para el desarrollo de la aplicación.

ASP

Active Server Pages (ASP) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft). Para escribir páginas ASP se utiliza un lenguaje de scripts, que se colocan en la misma página Web junto con el código HTML. Comúnmente este lenguaje de scripts es Visual Basic Script, que deriva del conocido Visual Basic, aunque también se pueden escribir los scripts ASP en otro lenguaje: JScript, que deriva a su vez del conocido Javascript [4].

PHP

PHP es un lenguaje de script interpretado en el lado del servidor, es gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación, usado frecuentemente para la generación rápida y dinámica de páginas Web [56]. Su funcionamiento es sencillo. Al ser PHP un lenguaje que se ejecuta en el servidor no es

necesario que su navegador lo soporte, es independiente del navegador, sin embargo, para que sus páginas PHP funcionen el servidor donde están alojadas debe soportar PHP. Además, como PHP está dentro de la corriente de software libre ha recibido varias contribuciones a lo largo del tiempo por parte de otros desarrolladores.

JSP

Es una tecnología Java que permite generar contenido dinámico para Web, en forma de documentos HTML, Lenguaje Extensible de Marcado (XML, por sus siglas en inglés) o de otro tipo. Con JSP es posible crear aplicaciones Web que se ejecuten en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java [47]. En JSP se crean páginas de manera parecida a como se crean en ASP o PHP -otras dos tecnologías de servidor-. Se generan archivos con extensión .jsp que incluyen, dentro de la estructura de etiquetas HTML, las sentencias Java a ejecutar en el servidor.

Ajax

Ajax, acrónimo de Asynchronous JavaScript And XML (JavaScript Asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas o aplicaciones de Internet ricas (RIA, por sus siglas en inglés). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano [59]. Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. Es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Modelo de Objetos de Documentos (DOM, por sus siglas en inglés).

JavaScript

Javascript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página Web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento [16]. Su uso se basa fundamentalmente en la creación de efectos

especiales en las páginas y la definición de interactividades con el usuario. Se utiliza principalmente para crear páginas Web dinámicas. Técnicamente, 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.

Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. Incluye una combinación de características que lo hacen único y está siendo adoptado por multitud de fabricantes como herramienta básica para el desarrollo de aplicaciones comerciales de gran repercusión. Es un lenguaje de desarrollo de propósito general, y como tal es válido para realizar todo tipo de aplicaciones profesionales. Los programas “ejecutables”, creados por el compilador de Java, son independientes de la arquitectura. Se ejecutan indistintamente en una gran variedad de equipos con diferentes microprocesadores y sistemas operativos [60].

Actualmente es un lenguaje de programación público. Permite escribir Applets (pequeños programas que se insertan en una página HTML) y se ejecutan en el ordenador local. Se pueden escribir aplicaciones para intrarredes, aplicaciones cliente/servidor, aplicaciones distribuidas en redes locales y en Internet. Es fácil de aprender y está bien estructurado. Las aplicaciones son fiables. Puede controlarse su seguridad frente al acceso a recursos del sistema y es capaz de gestionar permisos y criptografía. Es independiente de la plataforma, lo que significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware.

Características principales

Simple: Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos, lo que facilita un rápido y fácil aprendizaje.

Orientado a objetos: Java trabaja con sus datos como objetos y con interfaces a esos objetos.

Distribuido: Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp.

Robusto: Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo.

Arquitectura neutral: Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará.

Seguro: Las aplicaciones de Java resultan extremadamente seguras, ya que no acceden a zonas delicadas de memoria o de sistema, con lo cual evitan la interacción de ciertos virus.

Portable: Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo.

Interpretado: El intérprete Java (sistema run-time) puede ejecutar directamente el código objeto. Enlazar (linkar) un programa, normalmente, consume menos recursos que compilarlo, por lo que los desarrolladores con Java pasarán más tiempo desarrollando y menos esperando por el ordenador.

Multithreaded: Java permite muchas actividades simultáneas en un programa. Los threads (a veces llamados, procesos ligeros), son básicamente pequeños procesos o piezas independientes de un gran proceso. Al estar los threads contruidos en el lenguaje, son más fáciles de usar y más robustos. Esto proporciona un mejor rendimiento interactivo y mejor comportamiento en tiempo real.

Dinámico: Java se beneficia todo lo posible de la tecnología orientada a objetos. No intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución. Las librerías nuevas o actualizadas no paralizan las aplicaciones actuales.

El diseño de Java, su robustez, el respaldo de la industria y su fácil portabilidad han hecho de Java uno de los lenguajes con un mayor crecimiento y amplitud de uso en distintos ámbitos de la industria de la informática. En la parte del servidor, es más extendido que nunca, desde la aparición de la especificación de Servlets y JSP. Esto supone un importante avance ya que la programación es sencilla, flexible y extensible. A partir de su expansión entre la comunidad de desarrolladores, estas tecnologías han dado paso a modelos de desarrollo mucho más elaborados con frameworks (pe: Struts, Webwork) que se superponen sobre los servlets y JSP para conseguir un entorno de trabajo mucho más poderoso y segmentado en

el que la especialización de roles sea posible (desarrolladores, diseñadores gráficos,...) y se facilite la reutilización y robustez de código.

1.3.5 Framework

Los lenguajes de programación Web, constituyen tecnologías evolucionadas para el desarrollo de software, no obstante existen herramientas que proporcionan soporte y una mayor facilidad para su utilización, como es el caso de los framework. Un framework es una estructura software compuesta de componentes para el desarrollo de una aplicación. Se puede considerar como una aplicación genérica incompleta y configurable a la que es posible añadirle las últimas piezas para construir una aplicación concreta. Representa un patrón para el desarrollo y/o la implementación [27].

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un framework Web, por tanto, es un conjunto de componentes (por ejemplo clases en Java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

Características

- ✓ Abstracción de Localizadores Universales de Recursos (URLs, por sus siglas en inglés) y sesiones: No es necesario manipular directamente las URLs ni las sesiones, el framework ya se encarga de hacerlo.
- ✓ Acceso a datos: Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en Bases de Datos, XML, etc.
- ✓ Controladores: existen una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.
- ✓ Autenticación y control de acceso: Incluyen mecanismos para la identificación de usuarios mediante inicio de sesión y contraseña, y permiten restringir el acceso a determinadas páginas a determinados usuarios.
- ✓ Internacionalización.

- ✓ Separación entre diseño y contenido [27].

ZK Framework

ZK es un framework de aplicaciones Web en Ajax basado en componentes y eventos, completamente en Java de código abierto que permite una rica interfaz de usuario para aplicaciones Web sin usar JavaScript y con poca programación. Permite centrar toda la lógica de programación en el lado del servidor [55].

El núcleo de ZK es un mecanismo conducido por eventos basado en Ajax, sustentado sobre 70 componentes del Lenguaje basado en XML para la interfaz de Usuario (XUL, por sus siglas en inglés) y 80 componentes XHTML, y un lenguaje de marcación para diseñar interfaces de usuario. Los programadores diseñan las páginas de su aplicación en componentes XUL/XHTML ricos en características, y los manipulan con eventos disparados por la actividad del usuario final.

ZK utiliza el acercamiento llamado centrado-en-el-servidor para la sincronización de componentes, y los códigos de Ajax son completamente transparentes para los desarrolladores de aplicaciones Web. Además de la programación basada en componentes y orientación a eventos, de manera similar a Swing, ZK soporta el Lenguaje de Marcación de Interfaz de Usuario ZK (ZUML, por sus siglas en inglés) para la definición de una potente interfaz de usuario. Permite a un desarrollador mezclar diferentes tipos de lenguaje de marcación, tales como el lenguaje XUL de Mozilla y XHTML, todos ellos en la misma página. Además posibilita a los desarrolladores embeber conjuntos de instrucciones (scripts) en lenguaje Java y usar expresiones del lenguaje de programación JSP para manipular los componentes y acceder a los datos. Permite escribir y manejar los eventos de la interfaz de usuario con Java, además de la lógica del negocio.

Ventajas de ZK Framework

La accesibilidad Web obedece a los componentes Ajax con componentes que ofrecen unas atractivas ventajas para crear y maximizar la satisfacción y la experiencia del usuario y la eficiencia del trabajo. ZK promueve la productividad de los desarrolladores permitiendo una manipulación directa de la capa de la interfaz de usuario, Base de Datos y recursos

empresariales. Con elementos compilados XUL y HTML, ZK protege de las tecnologías de puertas cerradas y propietarias. ZK también compila con JSP, Java Server Faces, Portlet y tecnologías Java 2 Enterprise Edition incluyendo la habilidad de integrarse con Entornos de Desarrollo Integrado (IDEs, por sus siglas en inglés) Java ya existentes.

Presenta una arquitectura modular, es seguro y protege a las aplicaciones empresariales de inyecciones de JavaScript/SQL maliciosas, además de la exposición a la capa cliente de la lógica de negocio y la exposición de los datos en tránsito. Soporta alta escalabilidad y disponibilidad. ZK es también compatible con el balanceo de carga soportado por los servidores de aplicaciones modernos.

1.3.6 Patrón Modelo Vista Controlador

El uso de framework para desarrollar aplicaciones, provee de buenas prácticas de diseño e implementación, que a su vez representan ventajas a la hora de elaborar un software. El patrón Modelo-Vista-Controlador (MVC) es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados: el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio; el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información y el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema [27].

Al utilizar el patrón MVC existe una clara separación entre los componentes del programa; lo cual permite implementarlos por separado. La conexión entre el modelo y sus vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación. La principal ventaja de esta separación reside en la facilidad para realizar cambios en la aplicación puesto que cuando se hace un cambio de Bases de Datos, programación o interfaz de usuario, solo se afecta uno de los componentes. Además, se puede modificar uno de los componentes sin tener que conocer cómo funcionan los otros.

1.3.7 Tecnologías de Servidores Web

La arquitectura cliente/servidor es la arquitectura de software que describe el procesamiento entre dos o más programas: una aplicación y un servicio soportante, o sea, el cliente y el

servidor. Un servidor Web está diseñado para transferir hipertextos, páginas Web o páginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. El servidor Web se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte del cliente y responde a estas peticiones adecuadamente, mediante una página Web que se exhibe en el navegador, o mostrando el respectivo mensaje si se detectó algún error.

Servicios de Información de Internet

Servicios de Información de Internet (IIS, por sus siglas en inglés) Es una serie de servicios para ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT, luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP incluye una versión limitada de IIS. Este servicio convierte a una computadora en un servidor de Internet o Intranet, es decir, que en las computadoras que tienen este servicio instalado, se pueden publicar páginas Web, tanto local como remotamente.

Servidor Web Apache

Apache es el servidor Web hecho por excelencia, su robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Es una muestra, al igual que el sistema operativo Linux, de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar [33]. Se puede ejecutar en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. Es una tecnología gratuita de código fuente abierto. Constituye un servidor altamente configurable de diseño modular, y es muy sencillo ampliar sus capacidades. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto. Apache permite la creación de ficheros de reportes (logs) a medida del administrador, de este modo es posible tener un mayor control sobre lo que sucede en el servidor.

1.3.8 Sistemas Gestores de Bases de Datos

Las tecnologías actuales para el desarrollo de sistemas informáticos proveen valiosas herramientas para almacenar información a fin de garantizar su perdurabilidad en el tiempo., entre las que se encuentran las Bases de Datos y otras que permiten manipular esta información, como los Sistemas Gestores de Bases de Datos (SGBD). Los SGBD son conjuntos de programas, procedimientos y lenguajes que brindan las herramientas necesarias para trabajar con una Base de Datos, asegurando su integridad, confidencialidad y seguridad. Incorporan una serie de funciones que permiten definir los registros, sus campos, sus relaciones, y actualización de los datos [5].

Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la Base de Datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, la prevención de redundancias e inconsistencias y el mantenimiento de la integridad. Entre los más utilizados en el mundo se encuentran: MySQL, Microsoft SQL Server y PostgreSQL.

MySQL

MySQL es un SGBD sencillo de usar y increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales. Es una Base de Datos relacional, o sea, un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una Base de Datos relacional se usa el lenguaje estándar de programación SQL. Es de código abierto. Constituye una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad.

Microsoft SQL Server

Microsoft SQL-Server es un Sistema Gestor de Bases De Datos Relacionales. Es una herramienta de servidor, lo que quiere decir que se instala y usa recursos del servidor para procesar, interpretar, ejecutar y devolver los resultados a aplicaciones cliente.

No es libre, pero es potente y compatible con Windows. Permite transformación de datos, replicación, tiene sistemas de notificación y buena seguridad, aunque potencialmente expuesto a los problemas de seguridad de Windows. Entre sus características se pueden destacar: soporte de transacciones, escalabilidad, estabilidad, seguridad, soporta procedimientos almacenados, incluye un potente entorno gráfico de administración, permite trabajar en modo cliente-servidor; además, permite administrar información de otros servidores de datos [23].

PostgreSQL

PostgreSQL es un SGBD relacional orientado a objetos y libre, publicado bajo la licencia Berkeley Software Distribution (BSD). Comenzó como un proyecto denominado Ingres en la Universidad Berkeley de California. Ingres fue más tarde desarrollado comercialmente por la Relational Technologies/Ingres Corporation. Este proyecto sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto [42].

Está ampliamente considerado como un SGBD de código abierto muy avanzado. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre. PostgreSQL posee transacciones, integridad referencial, vistas y multitud de funcionalidades, pero es un poco más lento y pesado que MySQL, aunque en las últimas versiones del mismo esto ha mejorado mucho. Con la aparición de nuevas versiones, los desarrolladores de PostgreSQL argumentaron que empezaba una nueva era: más rápido, más fiable.

Características y ventajas

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, transacciones, optimización de consultas, herencia, y arreglos. Soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario. Incluye características avanzadas tales como las uniones (joins). Posee integridad referencial, la cual es utilizada para garantizar la validez de los datos de la Base de Datos.

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido. Utiliza una arquitectura proceso-por-usuario cliente/servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL. En este caso se utiliza el cliente pgAdmin III.

1.3.9 Metodologías para el desarrollo de software

El desarrollo de software no es sin dudas una tarea fácil. Como resultado a este problema ha surgido una alternativa desde hace mucho: la metodología. Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería [14]. En general, todas intentan lograr mejores aplicaciones, un mejor proceso de desarrollo y llevar a cabo un proceso estándar en la organización.

1.3.9.1 Proceso Unificado de Rational

El Proceso Unificado de Rational es un producto comercial desarrollado y comercializado por Rational Software, una compañía de IBM. Es un proceso de desarrollo de software, y representa la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Constituye una infraestructura flexible que proporciona prácticas recomendadas probadas y una arquitectura configurable [32].

Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papeles que se desempeñan en un determinado momento). Está dirigido por los casos de uso, pues no son sólo una herramienta para especificar los requisitos del sistema sino que también estos guían su diseño, implementación y prueba. Presta especial atención al establecimiento temprano de una buena arquitectura. Es iterativo e incremental: el trabajo se divide en partes más pequeñas o mini proyectos durante todo el proceso de desarrollo.

Tiene 4 fases de desarrollo, las que representan la estructura del ciclo de vida del proceso de desarrollo. Dentro de las mismas se realizan varias iteraciones en número variable según el proyecto, y se hace un mayor o menor hincapié en las distintas actividades.

Fase de concepción: definir y acordar el alcance del proyecto, identificar los riesgos, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.

Fase de elaboración: seleccionar los casos de uso que permiten definir la arquitectura base del sistema y se desarrollarán en esta fase, realizar la especificación de los mismos y el primer análisis del dominio del problema, diseñar la solución preliminar.

Fase de construcción: completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

Fase de transición: asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

RUP aplica varias de las mejores prácticas en el desarrollo moderno de software en una forma que se adapta a un amplio rango de proyectos y organizaciones. Esta metodología permite que todos los integrantes de un equipo de trabajo, conozcan y compartan el proceso de desarrollo, una base de conocimientos y los distintos modelos de cómo desarrollar el software utilizando un lenguaje modelado común, el Lenguaje Unificado de Modelado.

Ventajas de utilizar RUP

- ✓ Reconoce que las necesidades del usuario y sus requerimientos no se pueden definir completamente al principio.
- ✓ Permite evaluar tempranamente los riesgos en lugar de descubrir problemas en la integración final del sistema.
- ✓ Reduce el costo del riesgo a los costos de un solo incremento.

- ✓ Acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan para obtener resultados claros a corto plazo.
- ✓ Distribuye la carga de trabajo a lo largo del tiempo del proyecto ya que todas las disciplinas colaboran en cada iteración.
- ✓ Facilita la reutilización del código teniendo en cuenta que se realizan revisiones en las primeras iteraciones lo cual además permite que se aprecien oportunidades de mejoras en el diseño.

Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado es utilizado por RUP para definir sus diagramas y modelos. Es la síntesis de varias notaciones desarrolladas para la Ingeniería de Software, el análisis orientado a objetos y los métodos de diseño de los últimos veinte años[31].

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos y describe la semántica esencial de lo que estos diagramas y símbolos significan. El resultado es un lenguaje de modelado para la especificación, visualización, construcción y documentación de artefactos de un sistema.

Provee una serie de ventajas dada su utilización. El sistema es diseñado antes de ser codificado, por lo que se conoce lo que se obtendrá por adelantado. Es tarea fácil descubrir el código que se podrá reutilizar y desarrollarlo de la manera óptima. Se ahorra tiempo en desarrollo al no incurrir en errores realizando un buen diseño del sistema. Si se incorporan nuevos desarrolladores al proyecto, los diagramas UML les permitirán hacerse rápidamente una idea del sistema.

1.3.9.2 Metodologías ágiles

Existen otras metodologías para el desarrollo de software, que también garantizan buenos resultados en producto final. Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, las metodologías ágiles emergen como una

posible respuesta para llenar ese vacío metodológico. Por estar especialmente orientadas para proyectos pequeños, las metodologías ágiles constituyen una solución a medida para ese entorno, aportando una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto [7].

Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo. Esto ha llevado hacia un interés creciente en las metodologías ágiles. Están revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales [38].

No obstante a las ventajas que pueden representar para el desarrollo de proyectos este tipo de metodologías, RUP es una metodología muy seria para el proceso de creación de software, exitosa, y ha demostrado ser muy eficiente. Además, por las características de la investigación y el tiempo de desarrollo, es la más adaptable para la realización de este trabajo.

1.4 Análisis crítico de las fuentes y bibliografía utilizada

La bibliografía encontrada sobre las tendencias y tecnologías actuales para el desarrollo de aplicaciones Web fue amplia y muy diversa, lo que ayudó a elegir las tecnologías a utilizar para el desarrollo de la propuesta de solución. Fueron encontrados un total de 117 documentos, con información muy actualizada y reciente, lo que permitió obtener variados conocimientos sobre las tecnologías más usadas en estos días y analizar sus ventajas y posibilidades. Esto permitió seleccionar las más adecuadas para desarrollar la aplicación Web Sistema para la gestión de información de la guardia obrera y estudiantil en la Universidad de Holguín “Oscar Lucero Moya”, a fin de lograr mejores resultados.

No fueron encontrados suficientes antecedentes de sistemas informáticos para la guardia obrera y estudiantil. Solo se encontró una bibliografía relacionada con un sistema para el control de la guardia obrera y estudiantil en la Universidad de Oriente. No fue encontrada mucha bibliografía referente a este tema, en general solo algunos artículos en periódicos

nacionales mencionaban aspectos importantes referentes al mismo. También fueron consultados algunos reglamentos para el servicio de guardia obrera y estudiantil dictados en los años 2006 y 2010.

En general la bibliografía sobre la guardia obrera y estudiantil es escasa, pues no existen muchos documentos que traten este tema. No obstante, existe mucha y muy variada información acerca de las tecnologías que rigen el desarrollo del mundo actual y sus procesos. Es posible analizarlas, y dadas sus características y ventajas, elegir las más apropiadas para lograr superiores resultados.

1.5 Conclusiones del capítulo

A partir del estudio realizado se puede llegar a las siguientes conclusiones:

- La gestión de información de la guardia obrera y estudiantil conlleva una serie de procesos de vital importancia para lograr una mayor eficiencia en la realización de esta tarea.
- No existen suficientes antecedentes de sistemas informáticos en esta área.
- ZK Framewok constituye una valiosa herramienta para la creación de aplicaciones Web, y facilita y agiliza el desarrollo de las mismas.
- La metodología RUP ofrece un diseño adaptable y su principal objetivo es asegurar la producción de software de alta calidad, que cumpla las necesidades de sus usuarios finales.

Capítulo 2 Descripción de la solución propuesta

En este capítulo se presenta la propuesta de solución al problema planteado en la presente investigación, tomando como base la metodología de desarrollo RUP.

A continuación se realiza la modelación del negocio del entorno donde se desarrolla el sistema. Se exponen los requerimientos funcionales y no funcionales de la aplicación, y los paquetes en que se divide la misma, a fin de un mayor entendimiento, así como los casos de uso correspondientes. Igualmente se muestra un conjunto de artefactos del diseño considerados muy relevantes en el desarrollo del proyecto. Se hace un estudio de sostenibilidad del producto donde se puede apreciar el impacto administrativo, socio-humanista, ambiental y tecnológico y para finalizar se brindan los resultados del procesamiento estadístico de las valoraciones de los expertos acerca de la propuesta que se ofrece en esta investigación.

2.1 Modelación del Negocio

El modelo de casos de uso del negocio describe los procesos del negocio y cómo se benefician e interactúan los socios y clientes en estos procesos [32]. A continuación se describen las reglas del negocio, los actores y trabajadores, el diagrama correspondiente a los casos de uso del negocio, que representa la interacción entre actores y cada caso de uso; la descripción textual de los casos de uso y sus correspondientes diagramas de actividades y diagramas de clases del modelo de objetos.

2.2 Reglas del negocio

“Las reglas del negocio son políticas del negocio que deben cumplirse, que definen o regulan algún aspecto del negocio” [32]. Estas son una parte importante en el desarrollo de cualquier aplicación. Permiten identificar las restricciones o reglas del negocio para evitar la realización de acciones no válidas. Seguidamente se muestran las reglas identificadas en esta investigación:

- La guardia obrera y estudiantil es de vital importancia y de carácter obligatorio para todos los trabajadores y estudiantes.

- Los Jefes de Departamentos y Áreas deben confeccionar al final de cada curso el potencial de sus trabajadores y enviarlo al departamento de Seguridad y Protección.
- El Jefe de Seguridad y Protección es el encargado de confeccionar el Plan de Guardia Obrera y Estudiantil para el siguiente curso.
- El Plan de Guardia Obrera y Estudiantil una vez entregado debe ser revisado y si existen cambios, errores o incoherencias, se deben informar rápidamente al departamento de Seguridad y Protección.
- En el Puesto de Mando de la universidad se controla diariamente la guardia obrera y estudiantil y la asistencia a la misma.
- Los Decanos deben entregar la distribución diaria de la guardia estudiantil al Puesto de Mando.
- Los Jefes de Departamentos y Áreas deben informar a los trabajadores la correspondiente planificación de la guardia obrera con antelación.
- En las reuniones de trabajo semanalmente se dan los resultados del comportamiento general de la guardia, y en la última semana del mes se hace un resumen del mismo.
- Los trabajadores y estudiantes que por alguna situación no pueden realizar su guardia correspondiente, deben informar con tiempo de antelación a los Jefes de Departamentos y Áreas y Coordinadores de Año respectivamente.
- La relación de los estudiantes que están de guardia diariamente debe informarse al Puesto de Mando 24 horas antes de la realización de la misma.
- La relación del potencial obrero contempla los siguientes datos: No de orden, Nombre y Apellidos, Sexo, Dpto. o Área, Sugerencias (AOG o Patrulla Fin de Semana) y Observaciones.
- El Plan de Guardia Obrera y Estudiantil respecto a la guardia obrera recoge la siguiente información: No de orden, Nombre y Apellidos, Área, Dpto., Fecha, Tipo de Servicio y Horario; y en el caso de la guardia estudiantil ofrece los datos Facultad y Fecha.
- El modelo de asistencia a la guardia recoge: No de orden, Nombre y Apellidos, Ubicación, Hora de Entrada, Hora de Salida y Evaluación.

2.3 Descripción de los procesos del negocio

A partir de las entrevistas realizadas al personal encargado de la guardia obrera y estudiantil en la universidad, y de un análisis de las principales actividades relacionadas con el control y ejecución de la misma, se han identificado tres procesos principales. Estos procesos no son los únicos que se realizan, sino que en general representan los más importantes. Estos son:

- Confeccionar Plan de Guardia Obrera y Estudiantil
- Organizar guardia estudiantil
- Registrar cumplimiento guardia obrera y estudiantil

2.3.1 Actores y trabajadores del negocio

Los actores del negocio son aquellas personas que interactúan con el negocio para beneficiarse de sus resultados. Los trabajadores del negocio actúan en el negocio efectuando una o varias actividades, interactuando con otros trabajadores y manipulando entidades. En la tabla 2.1 y 2.2 se encuentran los actores y trabajadores del negocio involucrados con los procesos del negocio identificados anteriormente.

Tabla 2.1 Actores del negocio

Actores del negocio	Justificación
Dirección Universidad	La Dirección Universidad inicia los procesos de negocio Confeccionar Plan de Guardia Obrera-Estudiantil , donde se organiza la distribución general de los tipos, horarios y frecuencias de guardias de los trabajadores, y el caso de uso Registrar cumplimiento guardia obrera y estudiantil , donde se registra el cumplimiento de la guardia en general.
Decano	El Decano inicia el caso de uso Organizar guardia estudiantil , donde se realiza la distribución de la guardia estudiantil según la matrícula de los estudiantes en las facultades.
Jefe de Departamento/Área	El Jefe de Departamento/Área es el encargado de las actividades y tareas que se desarrollan en el departamento, interviene en el proceso de negocio Confeccionar Plan de Guardia Obrera y

	Estudiantil , con el propósito de recibir la distribución de la guardia para proceder a la realización de la misma.
--	--

Tabla 2.2 Trabajadores del negocio

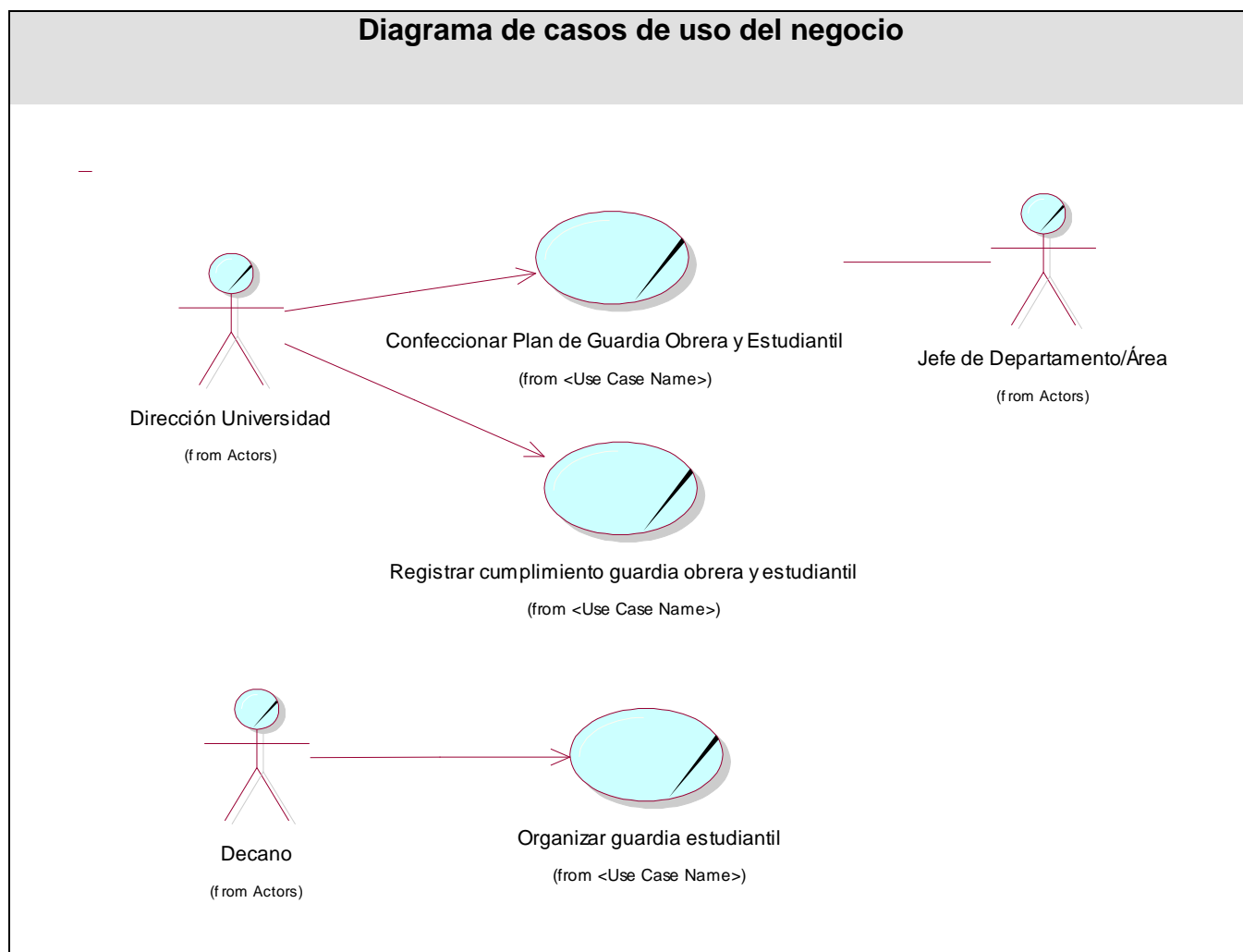
Trabajadores del negocio	Justificación
Jefe de Seguridad y Protección	El Jefe de Seguridad y Protección es el encargado de confeccionar el Plan de Guardia Obrera y Estudiantil, participa en los procesos de negocio Confeccionar plan de Guardia Obrera-Estudiantil y Organizar guardia estudiantil .
Puesto de Mando	El Puesto de Mando controla y registra el cumplimiento de la guardia obrera y estudiantil, participa en el proceso de negocio Registrar cumplimiento guardia obrera y estudiantil .

2.3.2 Diagrama de Casos de Uso del Negocio

Un caso de uso del negocio es un grupo de tareas lógicamente relacionadas que se llevan a cabo en una determinada secuencia tal que producen un resultado de valor. Cada caso de uso del negocio representa un proceso del negocio. El diagrama de casos de uso del negocio representa gráficamente los procesos del negocio y su interacción con los actores del mismo.

A continuación se muestra el diagrama correspondiente a los casos de uso del negocio:

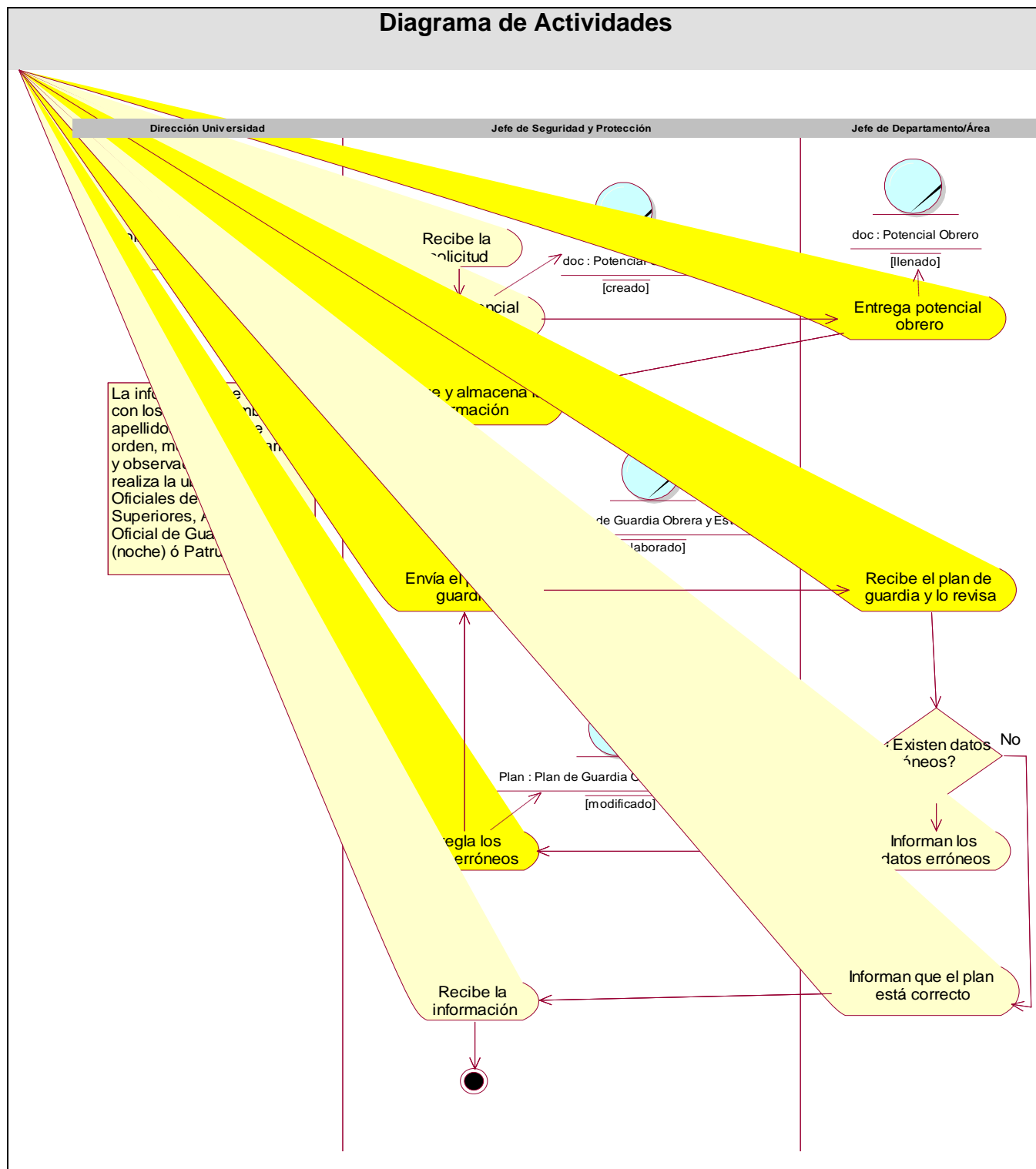
Figura 2.1 Diagrama de Casos de Uso del Negocio



2.3.3 Diagramas de actividades de los casos de uso del negocio

Los diagramas de actividades describen el flujo de trabajo asociado a los procesos de negocio. Muestran a través de las calles las responsabilidades de los actores y trabajadores del negocio, y a través del flujo de objetos cómo se utilizan las entidades del negocio. Las entidades del negocio representan los objetos que los trabajadores del negocio utilizan o generan durante la realización de los procesos de negocio. Las actividades que aparecen sombreadas en amarillo con una tonalidad más intensa son las actividades candidatas a informatizar en el sistema. En la figura 2.2 se muestra el diagrama de actividades del caso de uso **Confeccionar Plan de Guardia Obrera y Estudiantil**, el resto de los diagramas pueden ser consultados en la documentación digital adjunta.

Figura 2.2 Diagrama de actividad CUN Confeccionar Plan de Guardia Obrera y Estudiantil



2.3.4 Diagramas de clases del Modelo de Objeto

Los diagramas de clases del modelo de objetos describen el modelo de objetos del negocio, muestran la participación de los trabajadores y entidades del negocio y la relación que se establece entre ellos. Es muy importante que se construyan para cada proceso de negocio para una mejor claridad y comprensión. En la documentación digital adjunta pueden ser consultados los diagramas de clases del modelo de objeto correspondientes a cada caso de uso.

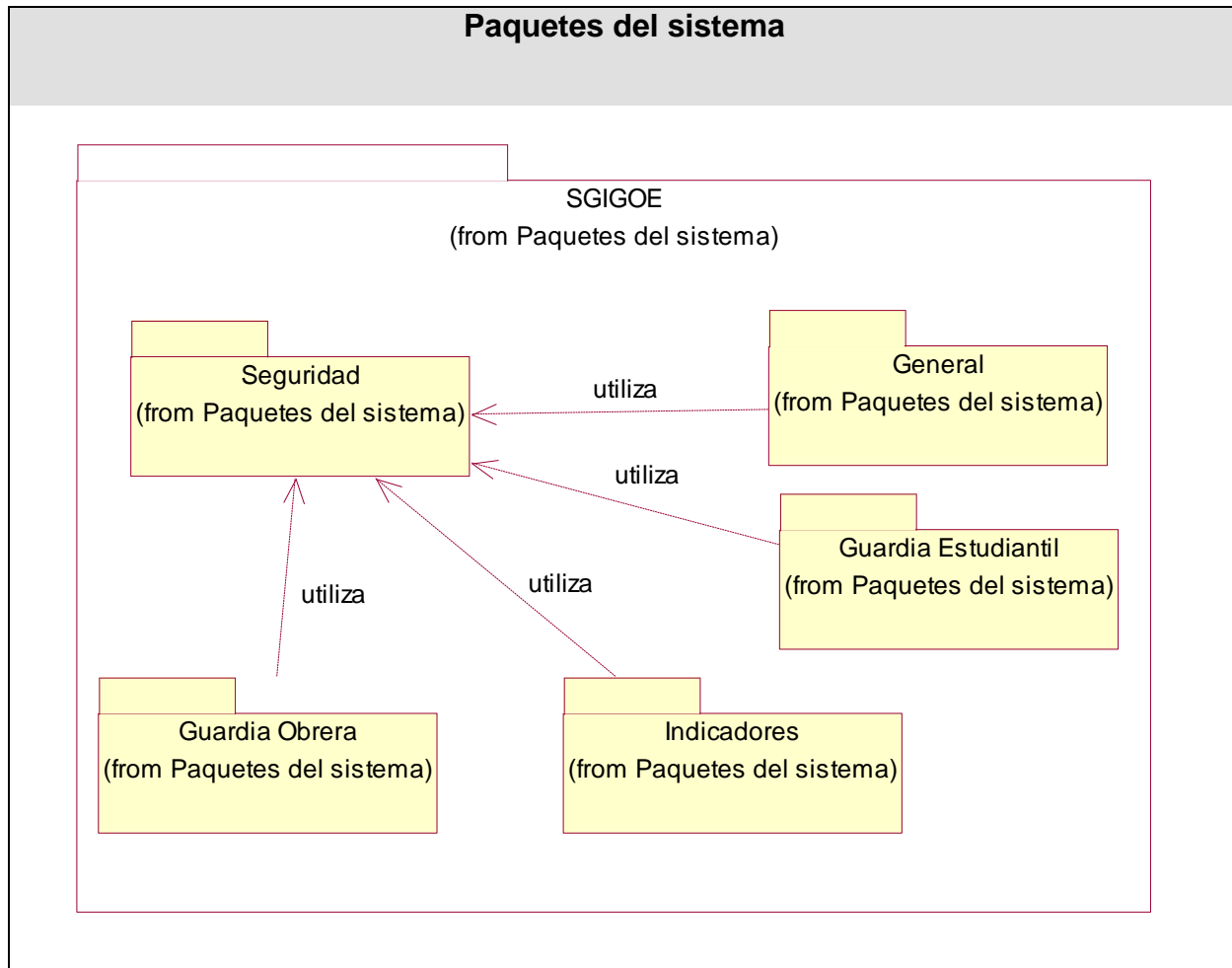
2.4 Modelación del sistema

En la modelación del sistema es donde se comienza a adentrarse en el software, sus funcionalidades y características externas e internas [32]. Este debe estar representado claramente por los casos de uso, los cuales describen el comportamiento del sistema desde el punto de vista del usuario, los actores del sistema y los requerimientos funcionales y no funcionales. A continuación se exponen los requerimientos funcionales de la solución propuesta, organizados en paquetes por las funcionalidades, los requerimientos no funcionales, actores y su justificación respecto a cada caso de uso, y los diagramas correspondientes a los casos de uso del sistema según su organización en paquetes.

2.4.1 Diagrama de paquetes del sistema

El diagrama que se muestra a continuación describe la división de la aplicación en los diferentes paquetes y las relaciones entre ellos.

Figura 2.3 Diagrama de paquetes del sistema



Los paquetes que conforman la aplicación tienen como objetivo principal agrupar de cierta manera las funcionalidades que la integran.

Paquete Seguridad: contiene las funcionalidades de actualización y validación de usuarios, iniciar y cerrar sesión.

Paquete Guardia Obrera: contiene las funcionalidades de actualización y visualización de los potenciales obreros y la distribución de la guardia obrera.

Paquete Indicadores: contiene las funcionalidades de actualización y visualización de los indicadores de la guardia obrera y estudiantil.

Paquete Guardia Estudiantil: contiene las funcionalidades de actualización y visualización de la distribución de la guardia estudiantil.

Paquete General: contiene las funcionalidades de registro y visualización del cumplimiento de la guardia, historiales de servicios de guardia y mostrar reportes de la guardia obrera y estudiantil.

2.4.2 Actores del sistema

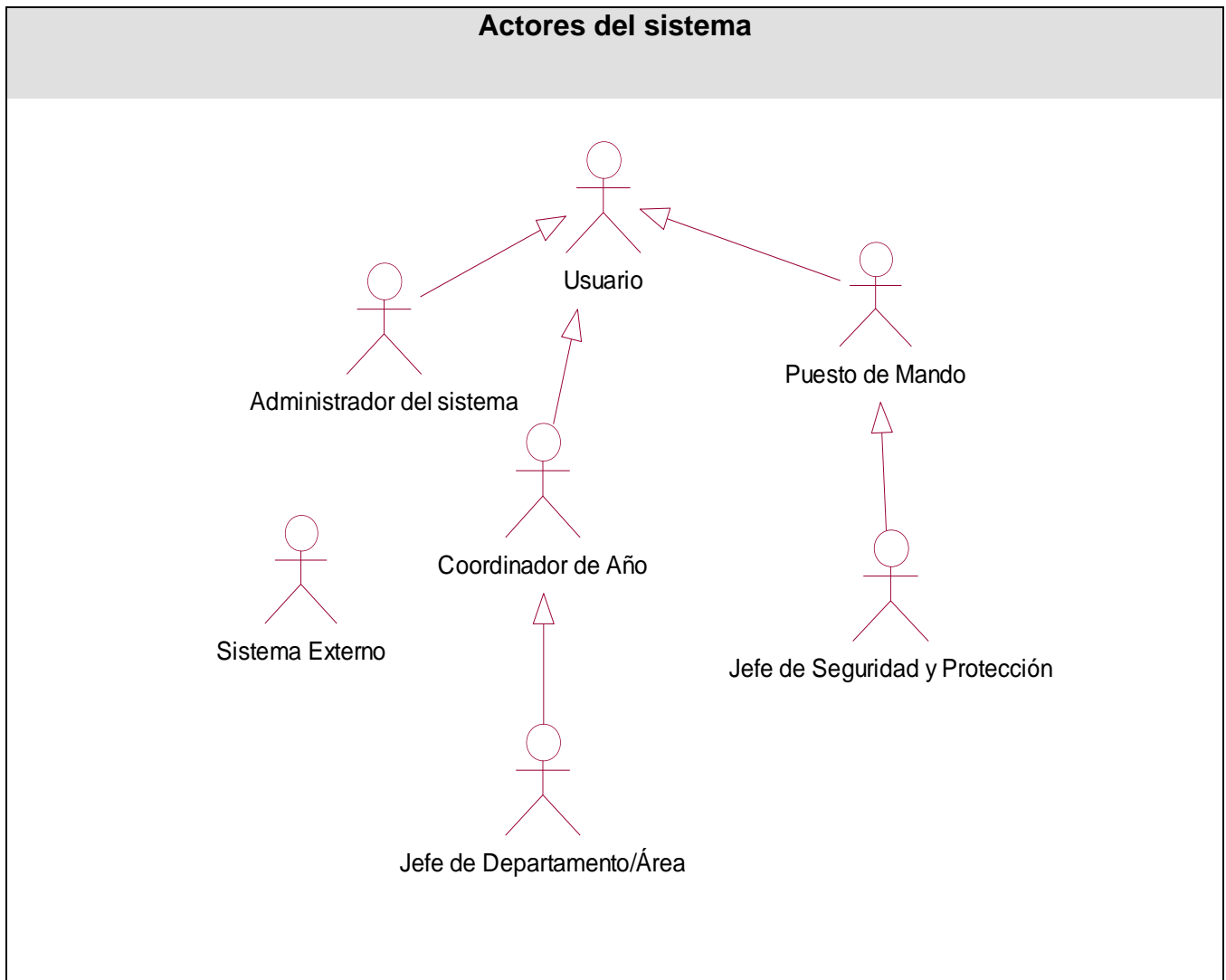
Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado, no se consideran parte del sistema, y pueden intercambiar información con él. Se destaca que los trabajadores del negocio que tienen actividades a automatizar son candidatos a actores del sistema, como son los casos del Jefe de Seguridad y Protección, y Puesto de Mando.

Tabla 2.3 Actores del sistema

Actores del sistema	Justificación
Usuario	Representa una generalización de los actores del sistema.
Administrador del sistema	Encargado de administrar las cuentas de usuario. Representa una especialización de Usuario.
Jefe de Seguridad y Protección	Encargado de la actualización de toda la información referente a la guardia obrera y la distribución de los días de guardia estudiantil. Representa una especialización de Puesto de Mando.
Jefe de Departamento/Área	Encargado de la actualización de toda la información referente al potencial obrero. Representa una especialización de Coordinador de Año.
Puesto de Mando	Encargado de la actualización del cumplimiento de la guardia en general. Representa una especialización de Usuario.
Coordinador de Año	Encargado de la distribución de la guardia estudiantil. Representa una especialización de Usuario.

A continuación en el diagrama de actores se muestra la estructura jerárquica entre ellos.

Figura 2.4 Actores del sistema



2.4.3 Requerimientos del sistema

Los requerimientos del sistema son las condiciones o capacidades que deben estar presentes en un sistema. Estos existen porque el tipo de producto en construcción exige cierta funcionalidad o calidad o porque el usuario quiere que el producto cumpla con ciertos requisitos. Los requerimientos pueden ser clasificados en requerimientos funcionales o requerimientos no funcionales.

2.4.3.1 Requerimientos funcionales

Los requerimientos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. Especifican acciones

que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física [32].

Los requerimientos funcionales que se enumeran a continuación han sido agrupados en los paquetes definidos anteriormente según los procesos que se ejecutan y las funcionalidades respecto a las necesidades del cliente.

Paquete Seguridad

R1 Validar Usuario

R2 Actualizar Cuenta de Usuario

R2.1 Insertar Usuario

R2.2 Eliminar Usuario

R2.3 Modificar Usuario

R3 Iniciar Sesión

R4 Cerrar Sesión

Paquete Guardia Obrera

R5 Actualizar Potencial Obrero

R5.1 Insertar trabajador

R5.2 Modificar trabajador

R5.3 Eliminar trabajador

R6 Visualizar Listado Potencial Obrero

R7 Visualizar Potencial Obrero

R8 Actualizar Guardia Obrera

R8.1 Insertar Guardia Obrera

R8.2 Modificar Guardia Obrera

R8.3 Eliminar Guardia Obrera

R9 Visualizar Listado Guardia Obrera

R10 Visualizar Guardia Obrera

Paquete Indicadores

R11 Actualizar Área

R11.1 Insertar Área

R11.2 Modificar Área

R11.3 Eliminar Área

R12 Visualizar Área

R13 Actualizar Departamento

R13.1 Insertar Departamento

R13.2 Modificar Departamento

R13.3 Eliminar Departamento

R14 Visualizar Departamento

R15 Actualizar Tipo de Servicio

R15.1 Insertar Tipo de Servicio

R15.2 Modificar Tipo de Servicio

R15.3 Eliminar Tipo de Servicio

R16 Visualizar Tipo de Servicio

R17 Actualizar Horario

R17.1 Insertar Horario

R17.2 Modificar Horario

R17.3 Eliminar Horario

R18 Visualizar Horario

R19 Actualizar Ubicación

R19.1 Insertar Ubicación

R19.2 Modificar Ubicación

R19.3 Eliminar Ubicación

R20 Visualizar Ubicación

Paquete Guardia Estudiantil

R21 Actualizar Guardia Estudiantil

R21.1 Insertar Guardia Estudiantil

R21.2 Modificar Guardia Estudiantil

R21.3 Eliminar Guardia Estudiantil

R22 Visualizar Listado Guardia Estudiantil

R23 Visualizar Guardia Estudiantil

Paquete General

R24 Insertar cumplimiento guardia obrera y estudiantil

R25 Visualizar cumplimiento guardia obrera y estudiantil

R26 Visualizar listado de servicios de guardia por diferentes criterios

R27 Visualizar reportes guardia obrera y estudiantil

2.4.3.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener; estas características son las que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto.

Apariencia o interfaz externa

- El diseño de la interfaz externa debe realizarse de manera sencilla, con un gran reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones.
- Debe tener un modelo claro y fácil de utilizar por los usuarios, para que se sientan identificados, cómodos y confiados con el mismo.
- De no poderse ejecutar una acción por un usuario, se debe visualizar un mensaje de error que especifique por qué no se pudo ejecutar.
- El color predominante en el diseño debe ser el azul en tonos suaves y claros, para compatibilidad con los colores característicos del logotipo de la universidad.

Usabilidad y Navegabilidad

- Debe haber una facilidad de uso por usuarios de cualquier nivel, presentando las funcionalidades visibles en todo momento que deben facilitar la navegación. Esto es necesario debido a que existen personas que interactuarán con la aplicación y tienen algunos conocimientos del trabajo con aplicaciones Web, pero existen otras que no los poseen.

- Debe brindar facilidades de selección de listas predeterminadas, así como búsquedas automatizadas y de autocompletamiento de los datos restantes.
- El diseño debe ser adaptable a diferentes resoluciones de pantalla.

Rendimiento

- La disponibilidad de trabajo en red contra el servidor debe ser constante.
- Disponibilidad las 24 horas del día haciendo posible su acceso en cualquier momento.

Soporte

- Se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra.

Portabilidad

- El producto debe poder ser usado bajo los Sistemas Operativos Windows, Linux y Unix.

Seguridad

- Debe posibilitar la creación de usuarios para el acceso a la aplicación.
- Debe validar el ingreso a la aplicación de los usuarios.
- Debe restringir el acceso a las funcionalidades de acuerdo con los roles jugados por los actores del sistema, ya que se puede incurrir en el desvío, pérdida, robo de información, entre otros problemas con la confidencialidad y fiabilidad de la misma.
- Se debe hacer una copia de seguridad de la base de datos de la aplicación.

Los requerimientos funcionales generados por los requerimientos no funcionales de seguridad son R1-R4.

Ayuda y documentos en línea

- Contar con una Ayuda que especifique el funcionamiento de la aplicación y tenga ejemplos claros y fáciles de entender, y un Manual de Usuario, para guiar al usuario respecto en la navegación de la aplicación.

Restricciones de diseño e implementación

- Debe ser una aplicación Web desarrollada con la utilización del Framework ZK, el lenguaje de programación Java y Sistema Gestor de Base de Datos en PostgreSQL.

Software

Se debe contar con los siguientes programas:

- Servidor de Base de Datos PostgreSQL versión 8.4, pues soporta grandes volúmenes de datos y tiene una buena velocidad de procesamiento.
- Máquina Virtual de Java (JDK 1.4 como mínimo).
- Servidor Web Apache Tomcat.
- En la máquina cliente se requiere un navegador capaz de interpretar JavaScript.

Hardware

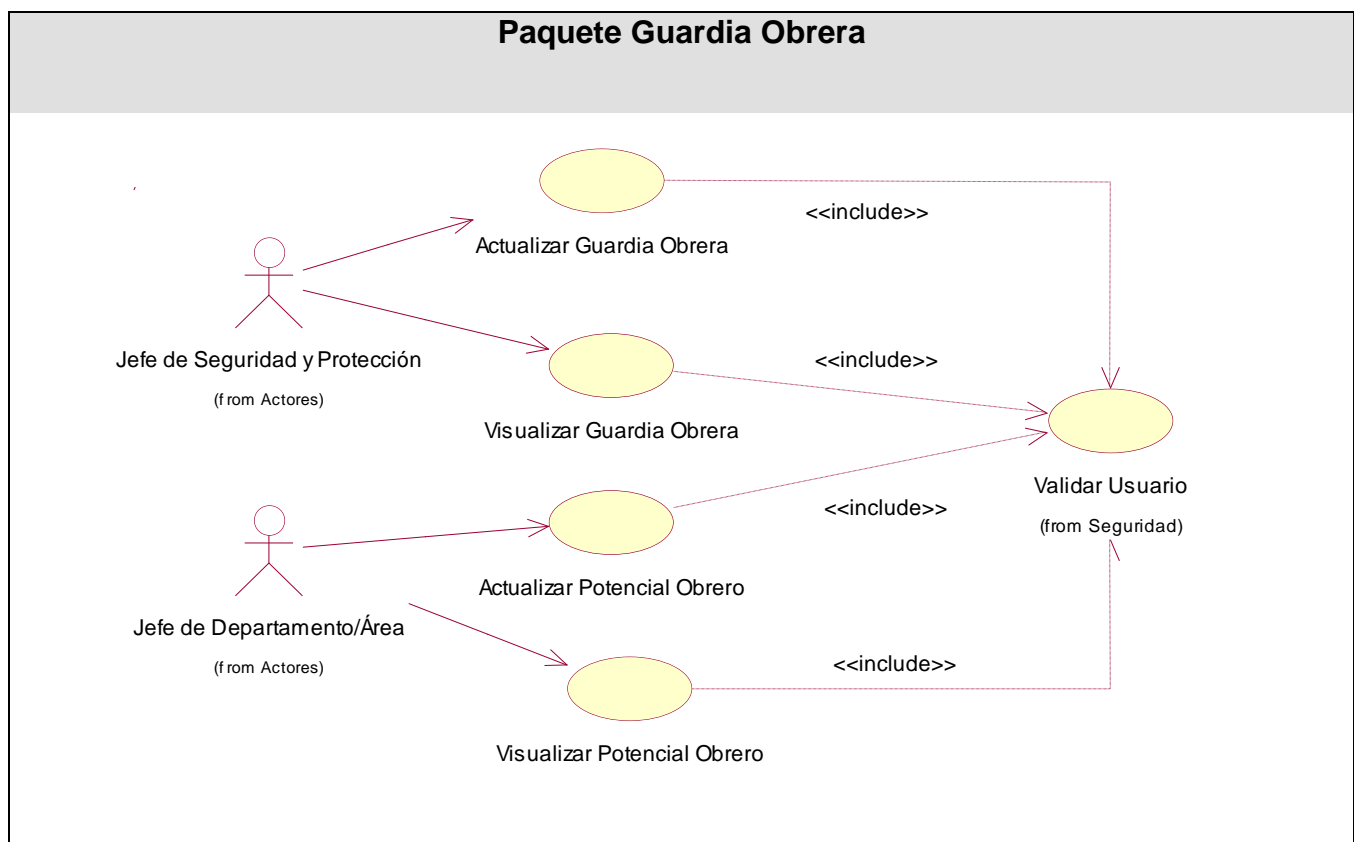
- El servidor debe tener como mínimo un procesador Pentium III, con 256 MB de memoria RAM.
- Todas las máquinas clientes deben tener un navegador instalado, preferentemente de la versión Internet Explorer 5.0 o superior, pueden ser además: Mozilla, Mozilla FireFox, Netscape y Opera (versiones recientes).
- Para el funcionamiento del sistema en el servidor será necesario el Sistema Operativo Windows 95 o superior, Linux o Unix, en sus versiones de S.O. servidores.
- Para el funcionamiento del sistema en las terminales cliente será necesario el Sistema Operativo Windows 95 o superior, Linux o Unix.

2.4.4 Diagrama de casos de uso del sistema

Los casos de uso del sistema describen el comportamiento del sistema desde el punto de vista del usuario. Un diagrama de casos de uso representa gráficamente a los procesos y su interacción con los actores.

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al Paquete Guardia Obrera.

Figura 2.5 Diagrama de casos de uso del sistema



2.5 Estudio de sostenibilidad

Al desarrollar e implantar un producto informático este trae consigo cambios positivos como negativos para los diferentes usuarios que interactuarán con él. Es preciso realizar un estudio

evaluativo de su impacto en las dimensiones ambiental, socio humanista, administrativa y tecnológica previsibles antes del diseño del proyecto, para de esta manera contribuir a que el producto desarrollado cumpla satisfactoriamente la necesidad que resuelve con un uso racional de recursos y la toma de decisiones adecuadas a las condiciones del contexto y el cliente. El estudio de este impacto está recogido en el estudio de sostenibilidad del proyecto.

La valoración de sostenibilidad de un producto informático es el “proceso de evaluación de impactos ambientales, socio-humanistas, administrativos y tecnológicos de un producto informático, previsibles desde el diseño del proyecto, que favorece su autorregulación, para la satisfacción de la necesidad que resuelve, con un uso racional de recursos y la toma de decisiones adecuadas a las condiciones del contexto y el cliente”[22]. En esta investigación se utilizó el Procedimiento para la valoración de sostenibilidad de un Producto Informático el cual tiene cuatro dimensiones de gestión de sostenibilidad: administrativa, socio-humanista, ambiental y tecnológica.

2.5.1 Factibilidad administrativa, técnica, ambiental y sociocultural

A continuación se muestra la valoración de sostenibilidad del sistema de gestión de información de la guardia obrera y estudiantil en la Universidad de Holguín “Oscar Lucero Moya” en las dimensiones administrativa, socio-humanista, ambiental y tecnológica.

Dimensión administrativa:

La dimensión administrativa incluye aspectos como: ahorro, gastos, calidad de la producción y los servicios, administración de recursos, toma de decisiones administrativas, entre otros.

En el estudio de sostenibilidad se hace un estudio de factibilidad del proyecto. Para ello se utiliza el Modelo Constructivo de Costos II (COCOMO II, por sus siglas en inglés). COCOMO II, es una herramienta utilizada para la estimación de algunos parámetros (costes en personas, tiempo, etc.) en el diseño y construcción de programas, y de la documentación asociada requerida para desarrollarlos, operarlos y mantenerlos, es decir, en la aplicación práctica de la Ingeniería del Software [45].

A continuación se muestra el estudio de factibilidad del Sistema para la gestión de información de la guardia obrera y estudiantil a partir de la utilización de COCOMO II, según los aspectos que miden la factibilidad económica del proyecto.

Tabla 2.4 Entradas externas (EI)

Nombre de las entradas externas	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
Insertar trabajador	4	4	Medio
Modificar trabajador	4	4	Medio
Eliminar trabajador	4	4	Medio
Insertar guardia obrera	5	3	Medio
Modificar guardia obrera	5	3	Medio
Eliminar guardia obrera	5	3	Medio
Insertar guardia estudiantil	4	5	Medio
Modificar guardia estudiantil	4	5	Medio
Eliminar guardia estudiantil	4	5	Medio
Actualizar área	1	1	Bajo
Modificar área	1	1	Bajo
Eliminar área	1	1	Bajo
Actualizar departamento	2	1	Bajo
Modificar departamento	2	1	Bajo
Eliminar departamento	2	1	Bajo
Actualizar horario	1	1	Bajo
Modificar horario	1	1	Bajo
Eliminar horario	1	1	Bajo
Actualizar tipo de servicio	1	1	Bajo
Modificar tipo de servicio	1	1	Bajo
Eliminar tipo de servicio	1	1	Bajo
Actualizar ubicación	1	1	Bajo
Modificar ubicación	1	1	Bajo

Eliminar ubicación	1	1	Bajo
Actualizar cuentas de usuario	2	3	Bajo
Modificar cuentas de usuario	2	3	Bajo
Eliminar cuentas de usuario	2	3	Bajo
Insertar cumplimiento guardia obrera	2	3	Bajo
Insertar cumplimiento guardia estudiantil	2	3	Bajo
Insertar curso	1	1	Bajo
Modificar curso	1	1	Bajo
Eliminar curso	1	1	Bajo

Tabla 2.5 Salidas externas (EO)

Nombre de las salidas externas	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
Visualizar listado trabajadores	3	4	Bajo
Visualizar potencial obrero	4	4	Medio
Listar servicios guardia obrera	8	9	Alto
Listar servicios guardia estudiantil	5	7	Alto
Visualizar cumplimiento guardia obrera	9	7	Alto
Visualizar cumplimiento guardia estudiantil	6	7	Alto
Visualizar guardia estudiantil	3	5	Bajo
Visualizar guardia obrera	4	6	Alto
Mostrar reporte guardia obrera	5	5	Medio
Mostrar reporte guardia estudiantil	3	5	Bajo

Tabla 2.6 Peticiones (EQ)

Nombre de las peticiones	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
Visualizar área	1	1	Bajo
Visualizar ubicación	1	1	Bajo
Visualizar tipo de servicio	1	1	Bajo
Visualizar horario	1	1	Bajo
Visualizar cuenta de usuario	2	2	Bajo
Visualizar curso	1	1	Bajo

Tabla 2.7 Ficheros internos (ILF)

Ficheros internos	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
tb_usuario	1	4	Bajo
tb_permiso	1	1	Bajo
tb_guardia_estudiante	1	7	Bajo
tb_estudiante	1	5	Bajo
tb_tipo_servicio	1	2	Bajo
tb_ubicacion	1	2	Bajo
tb_horario	1	2	Bajo
tb_plan_guardia	1	2	Bajo
tb_guardia_obrera	1	8	Bajo
tb_trabajador	1	7	Bajo
tb_departamento	1	3	Bajo
tb_area	1	2	Bajo

Tabla 2.8 Puntos de función desajustados

Elementos	Bajos	x Peso	Medios	x Peso	Altos	x Peso	Subtotal
ILF	12	84	0	0	0	0	84
ELF	0	0	0	0	0	0	0
EI	23	87	9	12	0	0	105
EO	3	20	3	10	5	35	62
EQ	6	18	0	0	0	0	18
						Total	264

Tabla 2.9 Características

Características	Valor
Puntos de función desajustados	264
Lenguajes	Java (80%), SGML (20%)
Instrucciones fuente x puntos de lenguaje x lenguaje	Java:55, SGML: 58
Instrucciones fuente x lenguaje	Java: 11 616 SGML: 3062,4
Total de instrucciones fuente	14,6784
Reducción 20% (miles de instrucciones fuentes)	11,7427

Se realiza una reducción de código fuente de aproximadamente un 20% en el diseño de las pantallas.

2.5.1.1 Cálculo del esfuerzo y tiempo de desarrollo

El esfuerzo asociado al desarrollo del sistema se denomina PM. Para estimar el esfuerzo se utilizan una serie de multiplicadores de esfuerzo y factores de escala con valores propios de acuerdo con las características del sistema. Se utilizan además otros valores calibrados. En las tablas 2.12 y 2.13 se muestran estos valores.

Tabla 2.10 Multiplicadores de esfuerzo

Multiplicadores de Esfuerzo	Descripción	Valor
RELY	Confiabilidad	3
DATA	Tamaño de la BD	3
CPLX	Complejidad del Producto	4
DOCU	Necesidades de Documentación	4
RUSE	Reutilización	3
TIME	Tiempo de Ejecución	4
STOR	Almacenamiento	3
PVOL	Volatilidad de la plataforma	4
ACAP	Capacidad de los analistas	5
PCAP	Capacidad de los programadores	3
PCON	Continuidad del Personal	2
APEX	Experiencia de los Analistas	4
PLEX	Experiencia con la plataforma	2
LTEX	Experiencia con Lenguajes y Herramientas	3
TOOL	Uso de Herramientas de SW	4
SITE	Desarrollo en diferentes Sitios	3
SCED	Requerimientos de Cronograma	3

Tabla 2.11 Factores de escala

Factores de escala	Descripción	Valor
PREC	Precedencia	4.96
FLEX	Flexibilidad	5.07
RESL	Riesgos	2.83
TEAM	Cohesión del equipo	0
PMAT	Madurez de las capacidades	3.12

Valores calibrados utilizados:

A = 2,94; B = 0,91; C = 3,67; D = 0,28

Cálculo del esfuerzo

El esfuerzo de desarrollo está representado por las siglas PM y se expresa en hombres por mes. La fórmula para obtenerlo se muestra a continuación:

$$PM = A * Size^E \prod_{i=1}^n EM_i \quad \text{Donde} \quad E = B + 0.01 * \sum_{j=1}^5 SF_j$$

E = 1.0698

PM ≈ 43 hombres por mes.

Cálculo del tiempo de desarrollo (TDEV)

El tiempo de desarrollo se obtuvo a partir de la fórmula:

$$TDEV = C * PM^F \quad \text{Donde} \quad F = D + 0.22 * (E - B)$$

TDEV ≈ 12 meses.

El tiempo de desarrollo estimado es de aproximadamente 12 meses para 3 hombres.

Reestructurado para una persona con el mismo esfuerzo es de aproximadamente 43 meses.

2.5.1.2 Cálculo del costo

Costo total (CT)

El costo total es el resultado de multiplicar el costo de hombres por mes (CHM) y el tiempo de desarrollo. Por tanto:

SP = \$ 225 Salario Promedio de un trabajador.

CHM = cantidad de hombres * SP

CHM = \$ 225 El proyecto se realiza por una persona.

Costo total:

$$CT = CHM * TDEV$$

CT \approx \$ 9675.

El esfuerzo calculado obtenido es el asociado al desarrollo del sistema que es de 43 Hombres por mes, o sea, 43 personas se necesitan para poder realizar este proyecto en un mes. Para una persona el tiempo de desarrollo estimado para realizar es de 43 meses. El salario promedio en este caso es de \$225 (considerando que es el salario común de un trabajador). Luego de terminar los cálculos, el costo total de proyecto adquiere el valor de 9675 pesos.

En el caso de la confección del Sistema para la gestión de información de la guardia obrera y estudiantil, no habrá gastos por mano de obra, ya que será realizado por una estudiante de la propia universidad como parte de un trabajo de Diploma. Para su desarrollo no se incurrirá en gastos adicionales, ya que se utilizará equipamiento existente en la Universidad, que es ya utilizado en actividades afines, por lo que tampoco se incrementará el consumo de electricidad.

Se ahorrarán recursos una vez implantado, que son esencialmente materiales de oficina, ya que al emplearse medios computarizados, se reducirá considerablemente el consumo de papel, tinta, etc.

Se mejorará considerablemente la calidad de la gestión de información y permitirá que se realice de una forma más rápida, además de agilizar la obtención de la información actualizada, por otra parte, garantiza que la gestión de la misma sea confiable y fácil de efectuar.

No genera ingresos directamente a la entidad donde se realiza ya que su finalidad es el uso por parte de los propios trabajadores de la misma.

Se utiliza software libre para la elaboración del sistema, por lo que no existirán gastos por el uso de las herramientas para llevar a cabo el desarrollo del proyecto.

A partir de lo antes analizado y los beneficios que proporciona el sistema, se llega a la conclusión que este es sostenible desde la dimensión administrativa.

Dimensión socio-humanista:

En las condiciones de vida y de trabajo se proporciona una mejora cuantiosa en los locales involucrados, ya que habrá una mayor eficiencia en la gestión de la información y se minimizará el tiempo y la complejidad de esta.

Por otro lado, no trae consigo la disminución de empleo en la organización, por el contrario, favorece el trabajo de las personas que ya son empleadas, aunque no resulta preciso emplear más personal.

Para evitar rechazo por parte de los usuarios ante el cambio, el sistema se implantará de forma tal que el personal se pueda ir adaptando al mismo fácilmente, se confeccionará un manual de usuario donde se expliquen las características, funcionalidades y ventajas del sistema, y de ser necesario, se impartirán cursos para adiestrar al personal que lo utilizará.

Por las consideraciones anteriores, se llega a la conclusión que el sistema es sostenible desde la dimensión socio-humanista.

Dimensión ambiental:

Se ahorrarán recursos que provienen del medio ambiente como los materiales de oficina, principalmente el papel, no se contribuye de modo alguno al deterioro del medio ambiente por causa de contaminación por ruido, interferencia, etc., todo lo contrario, su implantación ayuda en gran medida al cuidado del mismo en las diferentes esferas antes mencionadas.

En la interfaz del sistema se emplean colores suaves y refrescantes, todos en las distintas gamas de azules, manteniendo un aspecto agradable a la vista y una interfaz amigable, el tipo y tamaño de texto son adecuados. La navegación de la aplicación resulta fácil y no complica el acceso a los principales recursos de la misma.

El sistema facilita el acceso rápido a la información por lo que no será extenso el tiempo necesario de intercambio con el usuario, lo que no provoca que existan riesgos de daños a la vista y la postura de los implicados, efectos provocados por un tiempo prolongado de trabajo sin las condiciones óptimas (correcta iluminación, protectores de pantalla, asientos cómodos y con medidas correctas, ubicación correcta de la computadora).

A partir del análisis anterior y las utilidades que trae consigo el sistema, se llega a la conclusión que este es sostenible desde la dimensión ambiental.

Dimensión tecnológica:

Se cuenta con la infraestructura electrónica para el funcionamiento del sistema, ya que se dispone de los equipos técnicos suficientes para su correcto funcionamiento, no obstante es necesario sumar a la red que conecta los departamentos el de Seguridad y Protección, para lograr una correcta explotación del sistema.

Se recomienda una estabilidad en cuanto al manejo de la electricidad y el equipamiento técnico para garantizar la consistencia del sistema, la pérdida de la conexión entre los distintos ordenadores atentaría contra la disponibilidad de la información ya que no se tendría acceso a la misma. Por cuanto, debe tenerse cuidado con el estado de la red al ser este el principal factor que hace vulnerable al sistema.

El producto informático brinda facilidades de navegación que posibilitan su uso por todos los trabajadores de la entidad, no siendo necesaria una elevada preparación para maniobrar con el mismo. Se cuenta con la documentación para garantizar su buen funcionamiento y mantenimiento.

El sistema permite adaptarse a cambios que no lo alejen de su funcionalidad principal, por lo que es posible su mantenimiento y actualización. Es multiplataforma.

A partir de lo analizado anteriormente se concluye que el sistema es sostenible desde la dimensión tecnológica.

Conclusiones generales

Después del análisis de sostenibilidad del producto según las dimensiones administrativa, socio – humanista, ambiental y la tecnológica, se llega a la conclusión de que la herramienta propuesta cumple con todos los requerimientos antes señalados, por lo que se podrá decir que constituirá un producto informático sostenible.

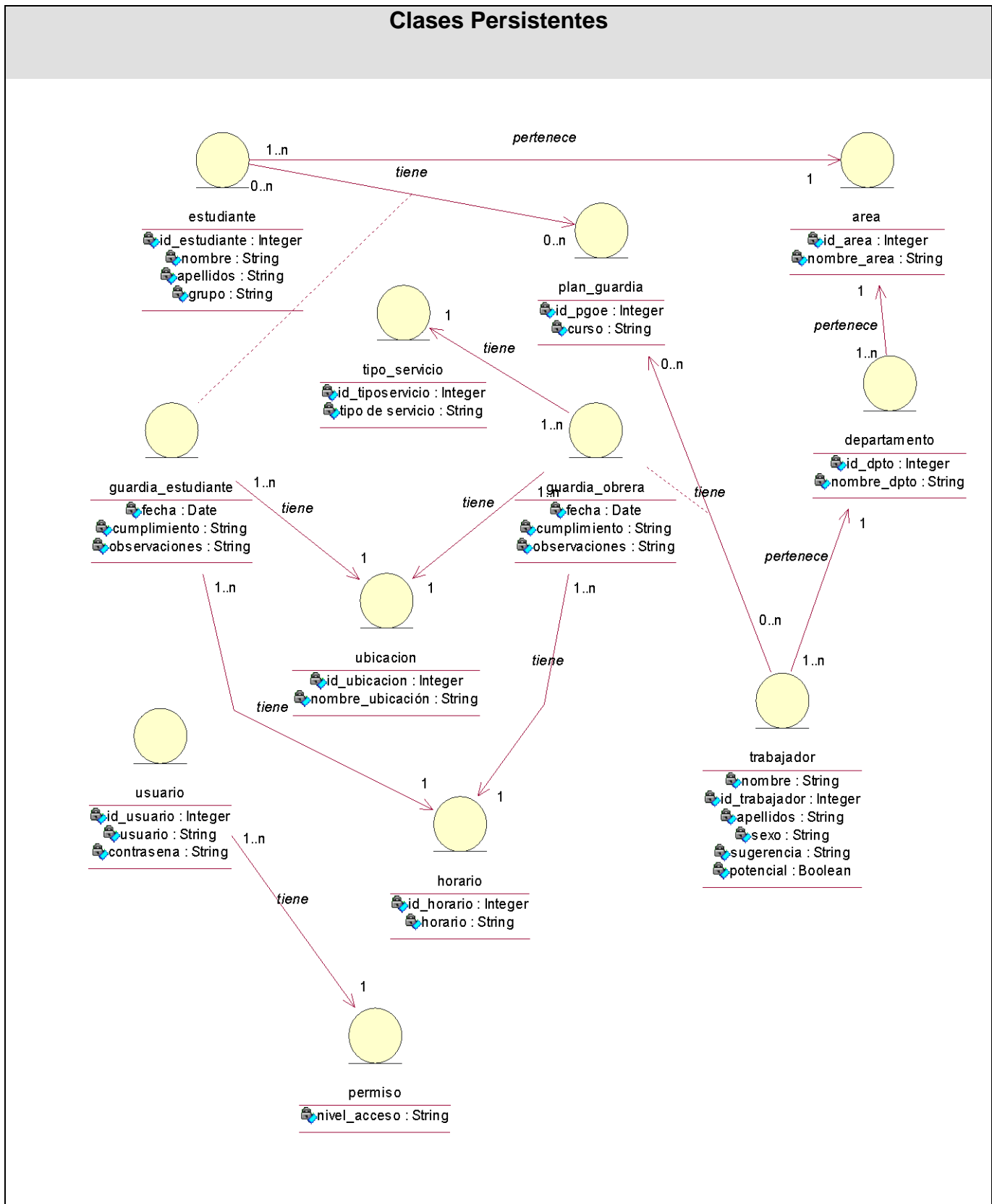
2.6 Base de Datos

2.6.1 Diagrama de Clases Persistentes y Modelo de Datos

Las clases persistentes representan la información que se almacenará de forma persistente en el sistema. Es importante, a partir del diagrama de clases, identificar las clases persistentes pues requieren una especial atención debido a las características de su almacenamiento físico. Las clases persistentes representan la información que el sistema necesita almacenar a largo plazo y para ello se definieron.

A continuación se muestra el diagrama de clases persistentes definido para la aplicación.

Figura 2.6 Diagrama de clases persistentes



El modelo de datos se utilizó para describir la estructura lógica de la información persistente almacenada por el sistema. Se creó a partir de las clases persistentes obtenidas previamente. Esto permitió mantener las estructuras que garantizaran la integridad, consistencia y confiabilidad de los datos. Fue empleado para definir la transformación de las clases persistentes en las estructuras de datos persistentes utilizadas en la aplicación Web.

El modelo de datos generado a partir del diagrama anterior puede consultarse en el Anexo 3.

2.6.2 Gestor de Base de datos

Las relaciones existentes entre tablas de la Base de Datos (BD) de la aplicación Web fueron permitidas por el gestor utilizado PostgreSQL. En el diseño de la BD incluye la definición de las tablas, los índices, las vistas, los procedimientos almacenados y otras estructuras específicas necesarias para almacenar, recuperar y eliminar la información persistente. Se debe asegurar que los datos persistentes se almacenen de forma consistente y eficientemente, definiendo el comportamiento que debe ser implementado en la BD.

La BD que contiene la información persistente, generada a partir del modelo de datos de la solución propuesta, se encuentra almacenada en el servidor de BD PostgreSQL, proporcionando grandes ventajas en cuanto a confiabilidad y rapidez en el acceso a los datos.

Para lograr la conexión a la BD del sistema se necesita una exitosa autenticación en el servidor de BD. La autenticación se realiza con un usuario que tiene los permisos de lectura y escritura sobre las tablas.

A través de las páginas dinámicas se puede establecer una conexión a la BD proporcionando el servidor, el usuario y la contraseña definidos en el servidor de BD. Una vez conectadas, se envía información a la BD y se escribe en las tablas o se leen datos desde estas para conformar los reportes que los usuarios solicitan.

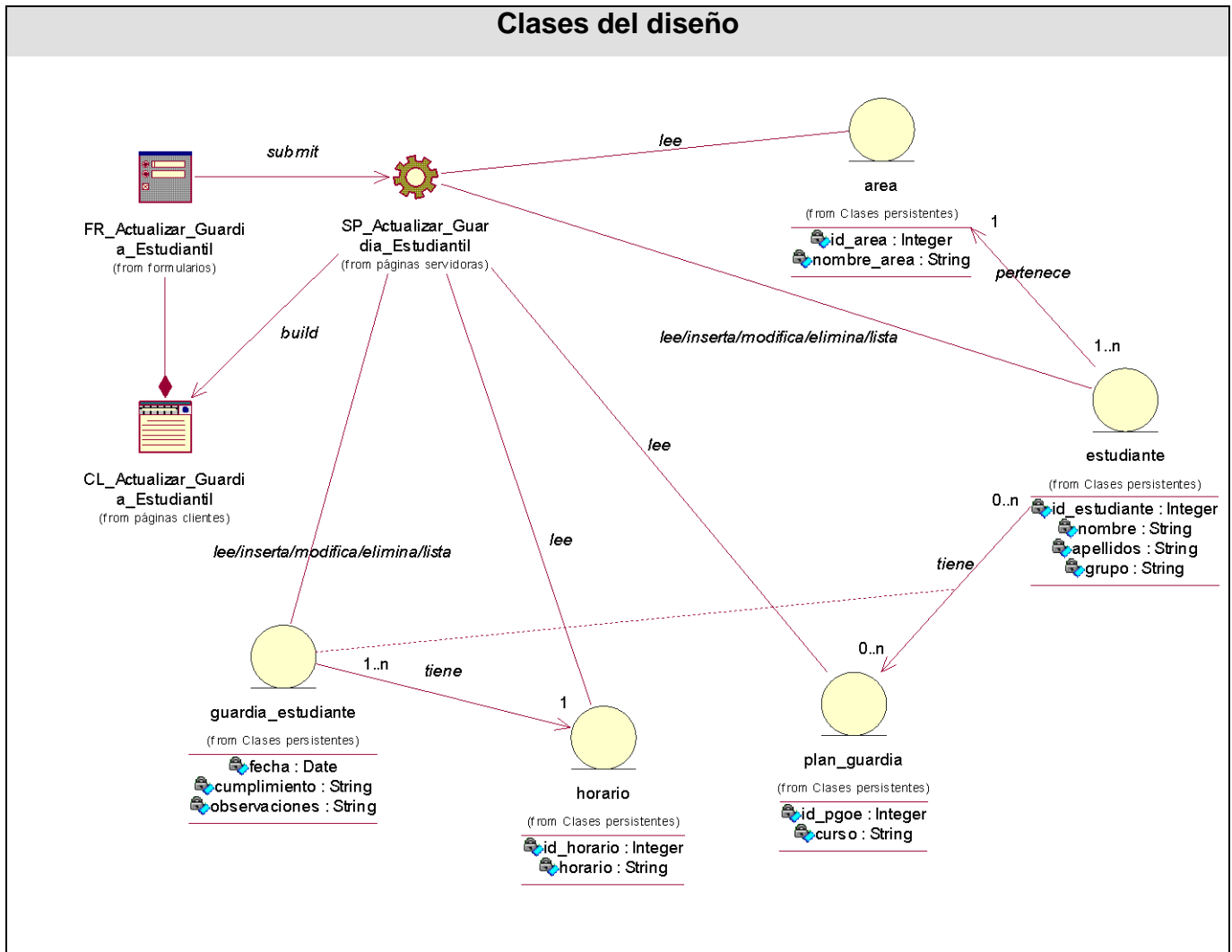
2.7 Diagrama de clases del diseño

Los diagramas de clases del diseño representan aquellas clases de larga vida dentro de la aplicación y las relaciones de estas con el resto de las clases. El diagrama de clases del diseño contiene clases, sus estructuras internas (atributos u operaciones) y relaciones con otras clases. Con estas características han sido especialmente útiles para representar qué se almacena dentro del sistema y su estructura.

Existen diferentes formas de nombrar a las clases; aquellas cuyo estereotipo es Client Page se nombran clases clientes, las clases cuyo estereotipo es Server Page se nombran clases servidoras, las clases cuyo estereotipo es HTML Form se nombran clases formularios y las clases cuyo estereotipo es Entity se nombran clases entidades. Para ayudar a identificar las clases se le añadió a los nombres de las clases clientes, servidoras y formularios los prefijos CL_, SP_ y FR_ respectivamente.

La clase cliente Principal representa la portada del sistema. Tiene enlaces a las clases servidoras que representan a cada uno de los módulos implementados. Estas clases servidoras crean las clases clientes que contienen los enlaces a la información que puede actualizarse. A continuación se muestra el diagrama de clases del diseño correspondiente al caso de uso Actualizar Guardia Estudiantil:

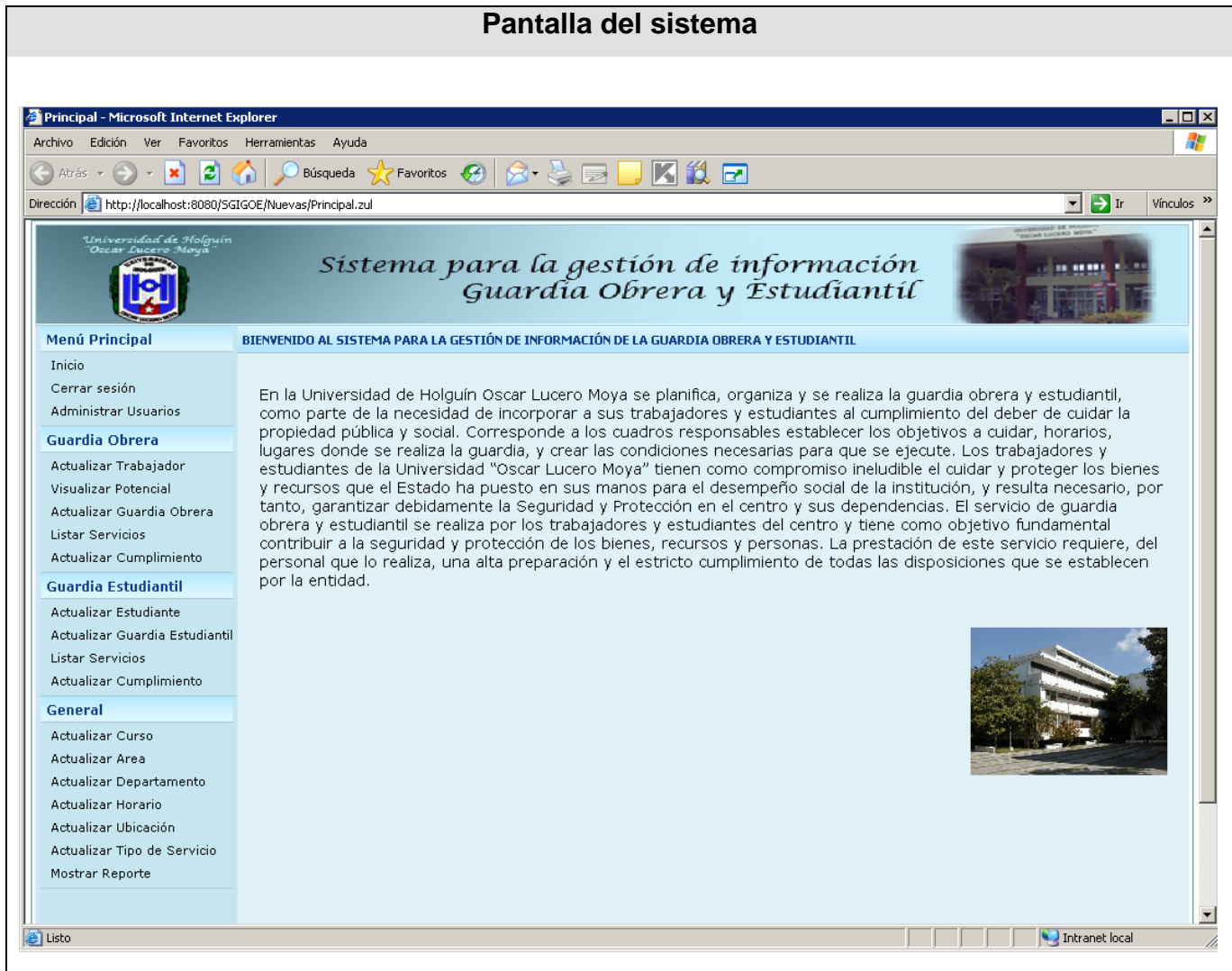
Figura 2.7 Diagrama de clases del diseño CUS Actualizar Guardia Estudiantil



2.8 Diseño Visual

Para el diseño de esta aplicación Web se emplearon algunos principios y técnicas de diseño visual en las páginas que la conforman. El diseño visual define la apariencia del sistema y es de gran importancia para lograr que el usuario se sienta satisfecho con la información que obtiene y con la forma en que lo hace. Los colores de la aplicación se encuentran en la gama de los azules claros, los cuales son suaves y refrescantes, a la vez que brindan una sensación de paz y tranquilidad al usuario. A continuación se muestra la pantalla de bienvenida de la aplicación, donde se puede apreciar el diseño visual de la misma.

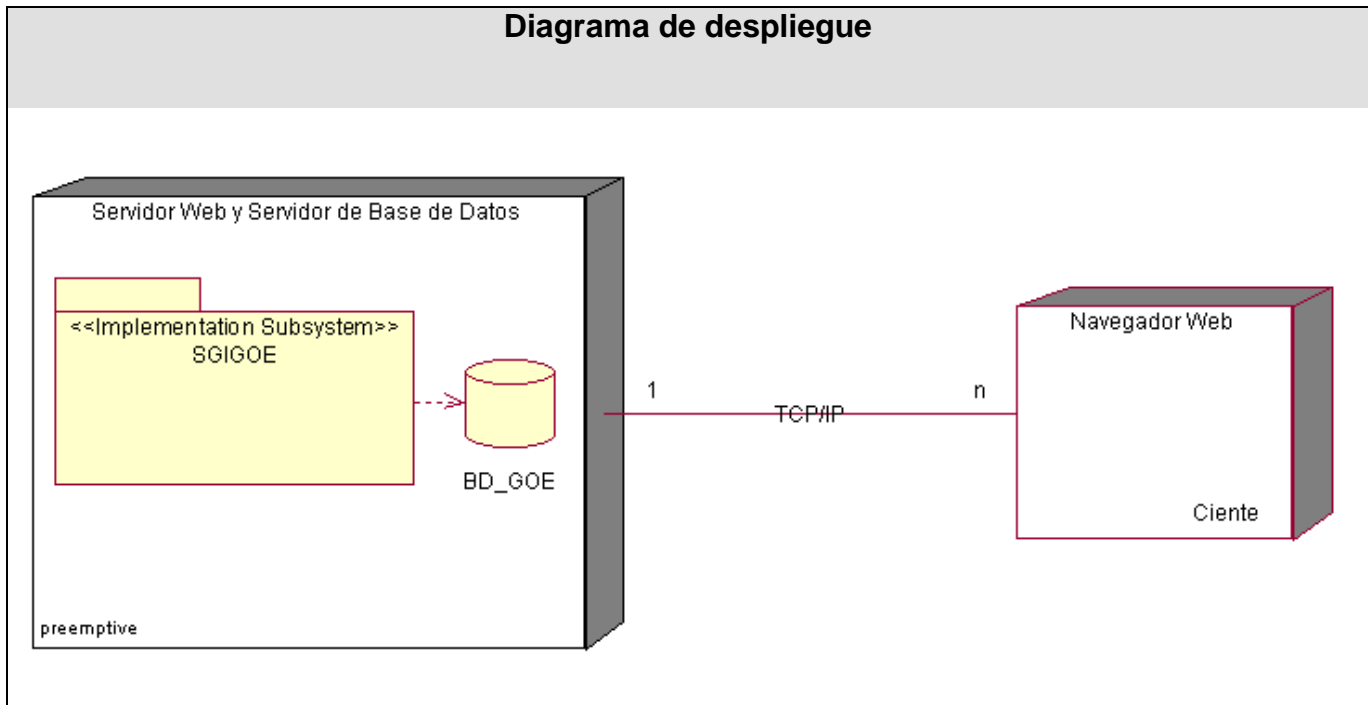
Figura 2.8 Pantalla de bienvenida de la aplicación



2.9 Diagrama de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño. En la figura 2.18 se puede observar el diagrama de despliegue para el sistema propuesto.

Figura 2.9 Diagrama de despliegue de la aplicación



El nodo nombrado servidor es una computadora con capacidad de procesamiento que se utiliza como servidor Web, donde se encuentran todos los componentes del sistema, y como servidor de BD, donde se almacenan los datos que son actualizados y consultados por los usuarios del sistema.

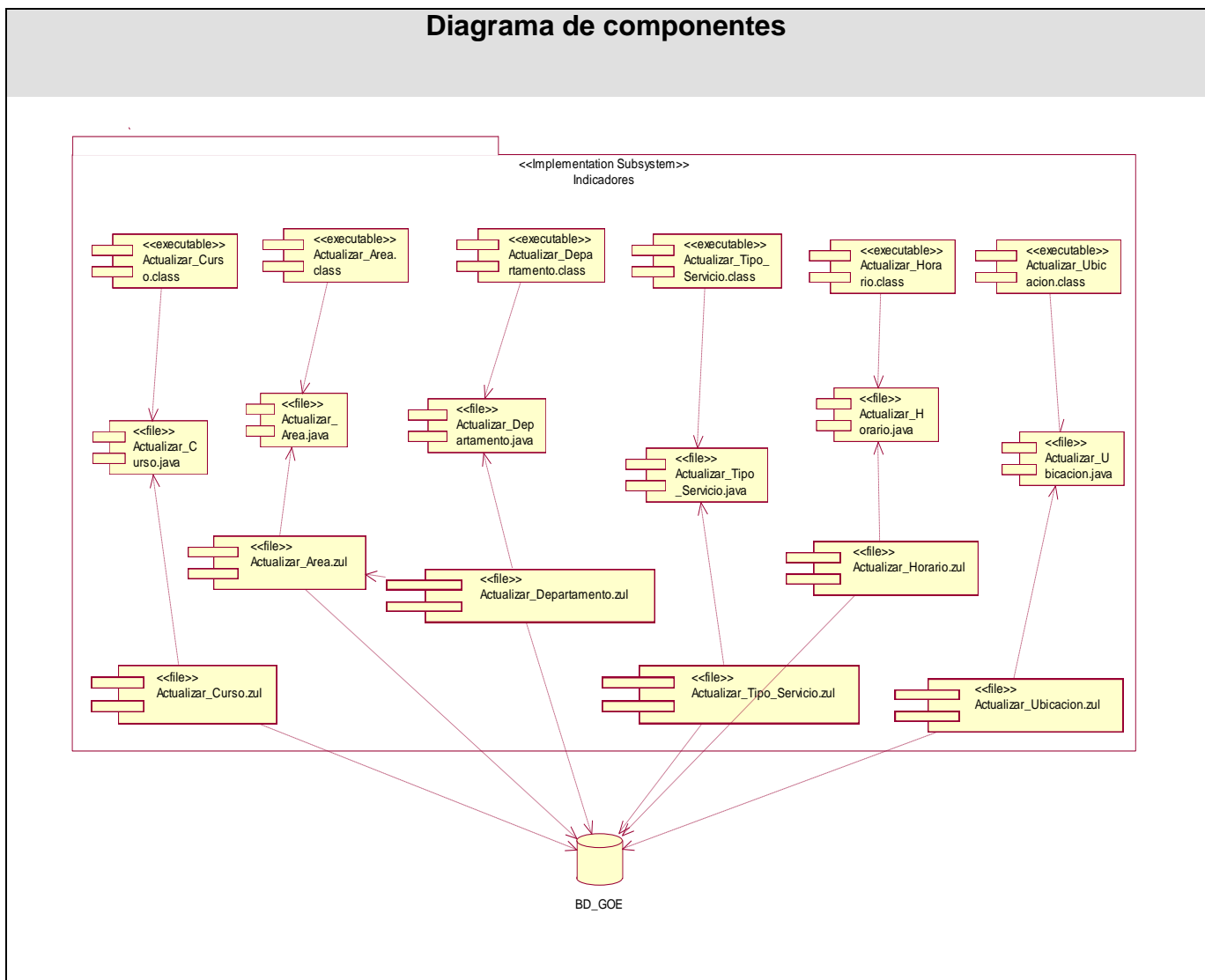
El nodo nombrado cliente representa el conjunto de computadoras a través de las cuales los usuarios pueden actualizar y consultar la información que se encuentra en el servidor. Para acceder al sistema, las computadoras clientes utilizan una aplicación cliente, por ejemplo, el navegador Mozilla Firefox. La comunicación entre las computadoras clientes y el servidor se establece con el uso del conjunto de protocolos de comunicación TCP/IP.

2.10 Diagrama de componentes

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño.

El diagrama de componentes muestra un conjunto de componentes y sus relaciones, que conforman un conjunto de interfaces y proporcionan la realización de esas interfaces. Para la modelación del sistema se desarrollaron varios diagramas de componentes. El caso de la aplicación, el framework utilizado presenta una filosofía de programación particular, por lo que surgen los componentes ".zul", los cuales conforman las páginas dinámicas de la misma. En la siguiente figura se muestra el diagrama correspondiente al paquete Indicadores y para ver el resto de los diagramas se puede consultar la documentación digital adjunta.

Figura 2.10 Diagrama de componentes Paquete Indicadores



2.11 Seguridad

La aplicación cuenta con un paquete de Seguridad donde se unifica la información que garantiza el acceso restringido de los usuarios al sistema. A cada usuario se le garantiza el acceso a la información del mismo en correspondencia al rol que desempeña con el objetivo de prever el desvío, pérdida, robo de información, entre otros problemas que se pueden presentar.

El usuario al registrarse en dependencia de su nivel de acceso realizará las tareas posibles. Si es Coordinador de Año solo podrá acceder a la parte de la guardia estudiantil, si es Jefe de Seguridad y Protección tendrá acceso a la parte de la guardia obrera y los indicadores además de otras funcionalidades, si es el usuario Puesto de Mando tendrá acceso al cumplimiento general además del Jefe de Seguridad y Protección, que es una especialización de Puesto de Mando; el administrador solo tendrá acceso a las cuentas de usuario, etc. En general, todos los roles son especializaciones de rol de usuario.

2.12 Integración con otras aplicaciones vinculadas a la gestión de información

En la Universidad de Holguín existen sistemas que permiten gestionar información de diferentes tipos. Han surgido a partir de las necesidades de llevar a cabo el registro y control de manera más eficiente de toda la información relacionada con los procesos que allí se ejecutan. Entre estos sistemas se encuentran el sistema “SIREH” correspondiente a la oficina de Recursos Humanos y “SIGENU” para la gestión de la información de los estudiantes, los cuales contienen información que puede ser utilizada en el sistema propuesto. De esta forma quienes trabajen con el mismo no tendrán la necesidad de introducir nuevamente la información común de estudiantes y trabajadores que se encuentre presente en los sistemas mencionados.

El sistema propuesto puede vincularse con estos otros sistemas mediante un Servicio Web, en este caso la información que es posible obtener es la de estudiantes, específicamente obtener datos de los estudiantes como son: nombre, apellidos y grupo; los cuales son necesarios para la distribución de la guardia estudiantil. El sistema “SIGENU” almacena la información en la base de datos “UHO”, utilizando el sistema gestor de base de datos PostgreSQL, el cual también es usado para guardar los datos de la aplicación propuesta. A

partir de una conexión con la base de datos “UHO”, es posible obtener de las tablas “student” y “xgroup” los datos de los estudiantes de todas las facultades de la universidad, y almacenarlos en la tabla “tb_estudiante” de la base de datos “BD_GOE” del sistema propuesto. Estos datos que están presentes en la base de datos de la aplicación son mostrados en la interfaz “Actualizar Guardia Estudiantil”, la cual lista los estudiantes por facultades para asignarles sus correspondientes guardias estudiantiles.

En general la integración del sistema propuesto con otras aplicaciones, garantiza la integridad y confiabilidad de la información, al lograrse la no redundancia de la misma y que esté presente de una sola forma para lo que se necesite.

En caso de que no se pueda cargar esta información el sistema “SGIGOE” está preparado para actualizar por parte del Coordinador de año y del Jefe de Departamento/Área, de forma que no se detenga el funcionamiento del mismo.

2.13 Aplicación del criterio de expertos

El método Delphi, considerado como uno de los métodos subjetivos de pronósticos más confiables, constituye un procedimiento para confeccionar un cuadro de la evolución estadística de las opiniones de expertos en un tema tratado. Permite rebasar el marco de las condiciones actuales más señaladas de un fenómeno y alcanzar una imagen integral y más amplia de su posible evolución, reflejando las valoraciones individuales de los expertos, las cuales podrán estar fundamentadas, tanto en un análisis estrictamente lógico como en su experiencia intuitiva. Para el caso de los usuarios sus criterios están fundamentados en el valor de uso que encuentran en una propuesta determinada [18].

El método Delphi fue creado en la década de los 60 del pasado siglo por Olaf Helmer y Dalkey Gordon, con el objetivo de elaborar pronósticos a largo plazo, referentes a posibles acontecimientos en varias ramas de la ciencia, la técnica y la política, y en el caso específico de los estudios de mercado.

En la presente investigación se aplicaron tres encuestas. La primera encuesta se aplicó para determinar el coeficiente de competencia del posible experto en el tema sistemas de gestión

de la información. Una vez seleccionados los expertos se procedió a aplicar la segunda encuesta. En la encuesta a los expertos fueron puestos a su consideración 19 aspectos del Sistema de gestión de información de la guardia obrera y estudiantil, que a juicio de la autora representan los más significativos y descriptivos. La tercera encuesta fue aplicada a los usuarios a fin de conocer sus opiniones respecto a las funcionalidades de la aplicación propuesta.

En la elaboración de las encuestas se incluyeron preguntas abiertas que les permitieran a los expertos y usuarios consultados exponer la capacidad de valoración del tema, debido a que constituye un elemento importante para derivar posteriores conclusiones, en cuanto a la eliminación, inclusión y cambio de denominación en algunos de los elementos analizados.

Fueron consultados diversos expertos en sistemas de gestión de información. Doctores, Másteres e investigadores en general, conocedores de este tipo de sistemas, fueron elegidos para aplicar las encuestas y arribar a conclusiones sobre el sistema propuesto.

Luego de hacer el procesamiento estadístico correspondiente se obtuvieron los siguientes resultados: en las encuestas a los expertos, estos coincidieron en que el 80 % de los aspectos consultados son muy relevantes, el 15 % de los aspectos consultados son bastante relevantes y el 5 % de los aspectos son relevantes para el apoyo en la gestión de la información. En la encuesta a los usuarios, estos coincidieron en que el 86 % de los aspectos consultados son muy relevantes, y el 14 % de los aspectos son bastante relevantes para el apoyo en la gestión de la información.

En la tabla 2.14 y 2.15 se muestran los aspectos que fueron considerados por los expertos y usuarios como muy relevantes para favorecer el proceso de gestión de la información.

Tabla 2.12 Aspectos considerados muy relevantes por los expertos consultados.

Aspectos
Intuitividad de interface Web
Metodología RUP + UML

Validación por criterio de expertos
Encuestas y entrevistas (requisitos)
Valoración de satisfacción - usuarios
Presentación de interface Web
Usabilidad
Actualizar potencial obrero
Actualizar guardia estudiantil
Actualizar guardia obrera
Visualizar cumplimiento guardia obrera
Visualizar cumplimiento guardia estudiantil
Realizar búsqueda por criterios
Mostrar reportes

Tabla 2.13 Aspectos considerados muy relevantes por los usuarios consultados.

Aspectos
Visualizar Potencial Obrero
Visualizar Guardia Obrera
Visualizar Guardia Estudiantil
Insertar cumplimiento guardia obrera y estudiantil
Visualizar listado de servicios de guardia por diferentes criterios

A partir de la aplicación del criterio de expertos por el método Delphi se obtuvieron resultados satisfactorios, ya que no hubo aspectos considerados poco o nada relevantes para el proceso de toma de decisiones. Por el contrario, todos los aspectos analizados estuvieron entre las categorías muy relevante, bastante relevante y relevante. Se puede concluir que el sistema propuesto efectivamente apoya la gestión de información de la guardia obrera y estudiantil en la universidad.

2.14 Conclusiones del capítulo

A partir del estudio realizado se puede llegar a las siguientes conclusiones:

- A partir del uso de la metodología RUP fue posible diseñar con eficiencia y claridad la propuesta para dar solución a la problemática existente, de manera que cumpliera con las necesidades de los usuarios.
- El Sistema para la gestión de información de la guardia obrera y estudiantil en su actual etapa de desarrollo, respalda el objetivo trazado en la investigación.
- El producto informático propuesto es sostenible desde las dimensiones administrativas, tecnológica, ambiental y socio-humanista.
- A partir de la aplicación del criterio de expertos se validó la hipótesis planteada para guiar la investigación, lo que quedó demostrado con los resultados obtenidos.

Conclusiones Generales

El Sistema de gestión de información de la guardia obrera y estudiantil es una nueva herramienta con la que sus trabajadores pueden contar para manipular su información en la Universidad de Holguín "Oscar Lucero Moya".

Con este sistema fue posible unificar la información necesaria para que el personal encargado del registro y control de la guardia obrera y estudiantil pueda gestionar la información de manera más eficiente. Se puede contar con la integridad y confiabilidad de la información sin limitar a los usuarios del sistema en su uso.

Es posible actualizar los datos desde cualquier lugar siempre que físicamente haya acceso a la información. Esto implica que el proceso, no solo de actualización de datos, sino también de su obtención, se haga de una forma más factible. El sistema obtenido es multiplataforma lo que permite que pueda seguirse utilizando si se cambia de Sistema Operativo. Fue desarrollado a partir de un estudio realizado sobre la guardia obrera y estudiantil en la Universidad de Holguín, de forma tal que sea sencillo para cuantos lo usen y que esté presente la información que realmente se necesita.

Se analizó la sostenibilidad del sistema propuesto y se arribó a la conclusión de que era sostenible desde las dimensiones administrativas, tecnológica, ambiental y socio-humanista, por lo que constituye un producto informático sostenible.

El sistema fue validado por medio de la aplicación del criterio de expertos por el método Delphi. Los expertos coincidieron en que el 80 % de los aspectos consultados son muy relevantes, el 15 % de los aspectos consultados son bastante relevantes y el 5 % de los aspectos son relevantes para el apoyo en la gestión de la información. Se puede concluir que el sistema propuesto efectivamente apoya el proceso de gestión de la información de la guardia obrera y estudiantil en la universidad dando cumplimiento a la hipótesis planteada.

Recomendaciones

A partir de los resultados y las experiencias derivadas de esta investigación se recomienda:

- Generalizar el sistema a otras universidades del país.
- Agregar nuevas funcionalidades al sistema como el envío de información por correo electrónico, entre otras.
- Implementar en la aplicación el servicio Web para la integración del Sistema para la gestión de información de la guardia obrera y estudiantil con “SIGENU” y “SIREH”.

Bibliografía

- [1] Aldana, Carlos Sáez. *Sistema de apoyo a la toma de decisiones en el proceso de negociación comercial* [Trabajo para optar por el título de Ingeniero en Informática]. Holguín: Universidad de Holguín; 2007.
- [2] Aldarias, Paco. *Sistema de Gestión de Base de Datos (SGBD): Postgres*. 2004.
- [3] Alvarez, Miguel Angel. *Qué es JSP*. [consultado:02 febrero 2010]. Disponible en: <http://www.desarrolloweb.com/articulos/831.php>. 2002.
- [4] Alvarez, Rubén. *Introducción a la programación en ASP*. [consultado:01 febrero 2010]. Disponible en: <http://www.desarrolloweb.com/articulos/244.php>.
- [5] Álvarez, Sara. *Sistemas gestores de bases de datos*. [consultado:01 feb 2010]. Disponible en: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>. 2007.
- [6] Amartino, Mariano. *Ajax: Un nuevo acercamiento a las aplicaciones Web*. [consultado:02 febrero 2010]. Disponible en: <http://www.maestrosdelweb.com/editorial/ajax/>. 2005.
- [7] Anaya, Adrian. *Desarrollo de software bajo metodologías ágiles (Agile Methods XP) en la práctica*. [consultado:12 febrero 2010]. Disponible en: <http://www.monografias.com/trabajos48/desarrollo-software-agil/desarrollo-software-agil.shtml>.
- [8] Bauer, Christian. *Hibernate: Object/Relational Persistence for idiomatic Java*; 2003.
- [9] Céspedes, Indira Cuba. *Sistema de gestión de información para la Residencia Universitaria de la Universidad de Holguín "Oscar Lucero Moya"* [Trabajo para optar por el título de Ingeniero en Informática]. Holguín: Universidad de holguín; 2009.
- [10] Cohen, Daniel. *Sistemas de información para la Toma de Desiciones*. Ed:McGRAW-HILL. 1999.
- [11] Conallen, Jim. *Building Web Applications with UML*. Ed:Addison-Wesley. 1999.
- [12] Cueva, Juan Manuel. *Introducción a UML: Lenguaje para modelar objetos*. 1999.
- [13] Darwin, Ian. *Java Cookbook*. Ed:O'Reilly. 2001.
- [14] Delgado, Erly. *Metodologías de desarrollo de software. ¿Cuál es el camino?* . [consultado:11 febrero 2010]. Disponible en: <http://www.monografias.com/trabajos60/metodologias-desarrollo-software/metodologias-desarrollo-software.shtml>.

- [15] Díaz, Luis Miguel. *¿Seguridad y Protección vs Calidad?* La Habana.
- [16] Eguíluz, Javier. *Introducción a JavaScript*. 2009.
- [17] Eguíluz, Javier *Introducción a Ajax*. 2008.
- [18] Expósito, Félix Rodríguez y García, Rita Concepción. *El método Delphi para el procesamiento de los resultados de encuestas a expertos o usuarios en estudios de mercado y en la investigación educativa*. 1999.
- [19] Fisher, Maydene; Ellis, Jon y Bruce, Jonathan. *JDBC™ API Tutorial and Reference, Third Edition*. Ed:Addison Wesley. 2003.
- [20] García, Aitor. *Guía del autoestopista a Hibernate*. 2003.
- [21] García, Rita Concepción. *La Gestión Ambiental de Proyectos Informáticos Sostenibles por estudiantes de Ingeniería Informática* [Tesis de Maestría en Gestión Ambiental]. Holguín: Universidad de Holguín; 2006.
- [22] García, Rita Concepción. *Procedimiento para la valoración de sostenibilidad de un Producto Informático*. 2006.
- [23] Gonzales, Mabel. *SQL SERVER*. [consultado:11 febrero 2010]. Disponible en: <http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml>.
- [24] González, Carlos D. *Curso Base de Datos PostgreSQL, SQL avanzado y PHP* [consultado:11 febrero 2010]. Disponible en: <http://www.usabilidadweb.com.ar/postgre.php> 2010.
- [25] González, Héctor Suárez. *Manual Hibernate*. 2003.
- [26] Greenspan, Jay y Bulger, Brad. *My SQL/PHP Database Applications*. Ed:M&T Books. 2001.
- [27] Gutiérrez., Javier J. *¿Qué es un framework web?*
- [28] Hernández, Rafael A. *Reflexiones sobre el uso del Software Libre en Cuba. Ventajas*. [consultado:01 febrero 2010]. Disponible en: <http://www.monografias.com/trabajos40/software-cuba/software-cuba.shtml?monosearch>. 2006.
- [29] Holzner, Steve. *Eclipse*. Ed:O'Reilly. 2004.
- [30] Horton, Ivor. *Beginning Java™ 2, JDK™ 5 Edition*. Ed:Wiley Publishing, Inc. 2005.
- [31] Jacobson, Ivar; Booch, Grady y Rumbaugh, James. *El Lenguaje Unificado de Modelado*. Ed:Addison Wesley. 2000.

- [32] Jacobson, Ivar; Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. Ed:Addison Wesley. 2000.
- [33] Kabir, Mohammed J. *Apache Server 2 Bible*. Ed:Hungry Minds, Inc. 2002.
- [34] Kon, Marcos D. *Software Libre*. [consultado:01 febrero 2010]. Disponible en: <http://www.Monografías.com/softwarelibre>.
- [35] Larman, Craig. *UML y Patronos*. Ed:Addison Wesley. 2000.
- [36] Lee, Susana. ***Participará la administración en la guardia obrera***. Granma 2007.
- [37] Letelier, Patricio. *Desarrollo de Software Orientado a Objeto usando UML*.
- [38] Letelier, Patricio. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*.
- [39] Malcolm, Graeme. *Programming Microsoft SQL Server 2000 with XML* Ed:Microsoft Press. 2001.
- [40] Martín, Carlos. *Desarrollo de aplicaciones basadas en tecnologías Web con lenguaje Java*. 2006.
- [41] Mendoza, María A. *Metodologías de Desarrollo de Software*. 2004.
- [42] Momjian, Bruce. *PostgreSQL: Introduction and Concepts* Ed:Addison-Wesley. 2001.
- [43] Morales, Einstein A. *La Educación y la Tecnología*. [consultado:15 enero 2010]. Disponible en: <http://www.monografias.com/trabajos22/educacion-y-tecnologia/educacion-y-tecnologia.shtml>.
- [44] Moreno, Ana María. *COCOMO II: Estimación de Proyectos Software*.
- [45] Moya, Antonio De La Fuente. *COCOMO v2 Modelo de estimación de costes para proyectos software* 1999.
- [46] Ortega, Ángel Luis Calvo. *Hibernate*; 2005.
- [47] Patzer, Andrew. *JSP: Ejemplos Prácticos*. Ed:Anaya Multimedia. 2002.
- [48] Polanco, Nelson Felipe. *Reglamento del Servicio de Guardia Obrero – Estudiantil*. 2010.
- [49] Preesman, Roger. *Ingeniería de software. Un enfoque práctico*. Ed:McGraw.Hill/Interamericana de España. 2002.
- [50] Quiñones, Ernesto. *Introducción a PostgreSQL*.
- [51] Rosario, Jimmy. *"TIC: Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual"*. [consultado:16 enero 2010]. Disponible en: <http://www.cibersociedad.net/archivo/articulo.php?art=221>. 2006.

- [52] Rosés, Francesc. *Introducción a Hibernate*. 2004.
- [53] Sekula, Andrzej. *La plataforma ZK*. Ed:Editorial MKM. 2008.
- [54] Sotomayor, Lucía. *Sistema para el Control de la Guardia Obrera y Estudiantil*. 2006.
- [55] Stäuble, Markus. *ZK Developer's Guide*. Ed:Packt Publishing Ltd. 2008.
- [56] Torre, Aníbal De La. *Lenguajes del lado servidor o cliente*. [consultado:01 febrero 2010]. Disponible en: http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.htm. 2006.
- [57] Valle, José Guillermo. *Definición arquitectura cliente servidor*. [consultado:21 enero 2010]. Disponible en: <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml> 2005.
- [58] Vegas, Jesús. *Introducción a las Aplicaciones Web*. [consultado:21 enero 2010]. Disponible en: <http://www.infor.uva.es/jvegas/cursos/buendia/pordocente/node11.html>.
- [59] Woychowsky, Edmond. *Ajax: Creating Web Pages with Asynchronous JavaScript and XML*. Ed:Prentice Hall. 2006.
- [60] Zukowski, John. *Programación Java 2 J2SE 1.4*. Ed:Anaya Multimedia. 2003.

Glosario de Términos

Herramientas Case: Las Herramientas de Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

HTML: Lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

HTTP: Protocolo de Transferencia de Hipertextos. Es el protocolo usado en cada transacción de la Web (WWW), es el modo de comunicación para solicitar páginas Web.

IDE: Entorno de Desarrollo Integrado. Es un programa compuesto por un conjunto de herramientas para un programador.

Multiplataforma: Término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Plataforma: Una plataforma es una combinación de hardware y software usada para ejecutar aplicaciones; en su forma más simple consiste únicamente de un sistema operativo, una arquitectura, o una combinación de ambos.

TCP/IP: Protocolo de Control de Transmisión/Protocolo de Internet, es el conjunto de protocolos que rigen las comunicaciones entre las computadoras en Internet.

Anexos

Anexo 1. Descripción textual de los casos de uso del negocio

Caso de uso del negocio	Confeccionar Plan de Guardia Obrera y Estudiantil
Actores	Dirección Universidad (inicia), Jefe de Departamento/Área.
Propósito	Confección del Plan de Guardia Obrera y Estudiantil.
Resumen	El caso de uso inicia cuando la Dirección Universidad solicita la confección al Jefe de Seguridad y Protección del Plan de Guardia Obrera y Estudiantil del siguiente curso, el caso de uso finaliza cuando el plan de guardia ha sido confeccionado, y revisado por el Jefe de Departamento/Área.
Casos de uso Asociados	-
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. La Dirección Universidad solicita al Jefe de Seguridad y Protección la confección del Plan de Guardia Obrera y Estudiantil para el siguiente curso.	2. El Jefe de Seguridad y Protección recibe la solicitud de confeccionar el Plan de Guardia Obrera-Estudiantil.
	3. El Jefe de Seguridad y Protección solicita el potencial obrero al Jefe de Departamento/Área.
4. El Jefe de Departamento/Área entrega el potencial obrero solicitado.	

	5. El Jefe de Seguridad y Protección recoge y almacena la información.
	6. El Jefe de Seguridad y Protección elabora el Plan de Guardia Obrera y Estudiantil.
	7. El Jefe de Seguridad y Protección envía el plan de guardia a los departamentos y áreas.
8. El Jefe de Departamento/Área recibe el plan de guardia y lo revisa.	
9. Si el plan no tiene datos erróneos, va al paso (11) si no, proceden de esta forma: a) informa los datos incorrectos al Jefe de Seguridad y Protección.	
	10. El Jefe de Seguridad y Protección arregla los datos incorrectos y va al paso (7).
11. El Jefe de Departamento/Área informa al Jefe de Seguridad y Protección que el plan está correcto.	
	12. El Jefe de Seguridad y Protección recibe la información.
Prioridad	Primario o crítico
Mejoras	La informatización de este proceso favorecerá la organización y el registro de la distribución de la guardia obrera y estudiantil de una forma estándar y sin dificultades, se evitarán pérdidas y posibles errores que se podrían presentar con el manejo inadecuado de estas informaciones.

Caso de uso del negocio	Registrar cumplimiento guardia obrera y estudiantil
Actores	Dirección Universidad (inicia)
Propósito	Registrar el cumplimiento de la guardia obrera y estudiantil.
Resumen	El caso de uso inicia cuando la Dirección Universidad solicita al Puesto de Mando el registro y control del cumplimiento de la guardia obrera y estudiantil. El caso de uso finaliza cuando el Puesto de Mando entrega los resultados del cumplimiento de la guardia.
Casos de uso Asociados	-
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. La Dirección Universidad solicita al Puesto de Mando el registro diario del cumplimiento de la guardia por parte de trabajadores y estudiantes.	2. El Puesto de Mando recibe la solicitud.
	3. El Puesto de Mando confecciona un modelo para el registro del cumplimiento.
	4. El Puesto de Mando llena el modelo con los trabajadores y estudiantes que están de guardia ese día.
	5. El Puesto de Mando registra los resultados del cumplimiento de la guardia por parte de trabajadores y estudiantes.
	6. El Puesto de Mando entrega los resultados del cumplimiento al departamento de Seguridad y

	Protección.
	7. El Jefe de Seguridad y Protección recibe los resultados del cumplimiento.
Prioridad	Primario o crítico
Mejoras	La informatización de este proceso favorecerá la organización y el registro del cumplimiento de la guardia obrera y estudiantil, además permitirá tener un mayor control del mismo.

Caso de uso del negocio	Organizar guardia estudiantil
Actores	Decano (inicia).
Propósito	Obtener la distribución de la guardia estudiantil.
Resumen	El caso de uso inicia cuando el Decano solicita al Jefe de Seguridad y Protección la distribución de los días de la guardia estudiantil. El caso de uso finaliza cuando el Jefe de Seguridad y Protección entrega la distribución al Decano.
Casos de uso Asociados	-
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El Decano solicita al Jefe de Seguridad y Protección la distribución de los días de la guardia estudiantil.	
2. El Decano entrega la matrícula de la facultad para el curso.	3. El Jefe de Seguridad y Protección recibe la solicitud y la matrícula de la facultad para la guardia estudiantil.
	4. El Jefe de Seguridad y Protección distribuye la guardia estudiantil asignando una cantidad determinada de días mensuales a cada facultad.
	6. El Jefe de Seguridad y Protección entrega la

	distribución de la guardia estudiantil al Decano.
7. El Decano recibe la distribución de la guardia estudiantil.	
Prioridad	Primario o crítico
Mejoras	La informatización de este proceso favorecerá la organización de la guardia estudiantil, con lo cual se logrará un mayor control de la misma y posibilitará la obtención de reportes.

Anexo 2. Descripción de los Casos de Uso del Sistema

Nombre del CUS:	Validar Usuario
Actores del sistema:	Es un caso de uso incluido en el resto de los casos de uso del sistema. El actor que lo inicia es el que inicie cualquiera de los casos de uso que lo incluyen.
Propósito:	Comprobar que la opción limitada accedida por el usuario se encuentre permitida.
Descripción: El caso de uso se inicia cuando el usuario vigente en el sistema realiza alguna acción. El sistema comprueba el nombre de usuario, contraseña y el nivel de acceso. De encontrarse correctamente, el sistema comprueba sus privilegios y, a partir de estos, se le permite acceder al lugar deseado por el usuario. En caso contrario, el sistema se va a la pantalla de autenticación. El caso de uso finaliza cuando el sistema comprueba si puede o no acceder a esa opción.	
Referencias:	R1.
Requerimientos especiales:	-

Nombre del CUS:	Actualizar Cuentas de Usuario
Actores del sistema:	Administrador (inicia).
Propósito:	Insertar, modificar o eliminar un usuario.
Descripción: El caso de uso se inicia cuando el Administrador decide actualizar las cuentas de usuario. Puede insertar un usuario, modificar o eliminar uno existente. El caso de uso finaliza cuando se accede a otras opciones o se cierra la aplicación.	
Referencias:	R1, R2, R2.1, R2.2, R2.3. Este caso de uso tiene incluido el de Validar usuario.

Precondiciones:	Para modificar o eliminar debe existir en la Base de Datos el usuario y para insertar uno nuevo deben estar definidos los niveles de acceso. El usuario debe estar autenticado como Administrador.
Poscondiciones:	Las cuentas de usuario quedan actualizadas.
Requerimientos especiales:	-

Nombre del CUS:	Iniciar sesión
Actores del sistema:	Usuario (inicia).
Propósito:	Autenticarse en el sistema.
Descripción: El caso de uso se inicia cuando el Usuario decide entrar al sistema. Se comprueba que el nombre del usuario y la contraseña se encuentren correctamente. De ser así, se parte a las opciones según el rol y nivel de acceso que tiene el usuario; de lo contrario, se retorna a la misma pantalla de autenticación. El caso de uso finaliza cuando el Usuario accede a las opciones que tiene permiso o cuando sea rechazada su entrada y retorne a la pantalla de autenticación.	
Referencias:	R1, R3. Este caso de uso tiene incluido el de Validar usuario.
Poscondiciones:	En caso de estar correcto los datos el usuario ha iniciado sesión satisfactoriamente.
Requerimientos especiales:	-

Nombre del CUS:	Cerrar sesión
Actores del sistema:	Usuario (inicia).
Propósito:	Cerrar sesión en el sistema.

Descripción: El caso de uso se inicia cuando Usuario decide cerrar la sesión en el sistema. El caso de uso finaliza cuando retorna a la pantalla de inicio de la aplicación.	
Referencias:	R4
Precondiciones:	El usuario debe estar autenticado en el sistema.
Poscondiciones:	El sistema ha eliminado la información del usuario autenticado.
Requerimientos especiales:	-

Nombre del CUS:	Actualizar potencial obrero
Actores del sistema:	Jefe de Departamento/Área. (Inicia)
Propósito:	Actualizar los datos del potencial obrero del departamento o área.
Descripción: El caso de uso inicia cuando el actor accede a la opción de insertar, eliminar, modificar el potencial obrero. Puede insertar un nuevo trabajador, modificar los datos de alguno ya existente en la Base de Datos o eliminar alguno. El caso de uso finaliza cuando el actor ha actualizado el potencial obrero correspondiente a su departamento o área y accede a otras opciones o cierra la aplicación.	
Referencias:	R1, R5, R5.1, R5.2, R5.3, R6. Este caso de uso tiene incluido el de Validar usuario.

Precondiciones:	Para modificar o eliminar debe existir en la Base de Datos el trabajador y para insertar uno nuevo deben estar definidos los datos del mismo como Área, Departamento y Tipo de Servicio. El usuario debe estar autenticado como Jefe de Departamento/Área.
Poscondiciones:	Los datos del potencial obrero quedaron actualizados.
Requerimientos especiales:	-

Nombre del CUS:	Visualizar potencial obrero.
Actores del sistema:	Jefe de Departamento/Área. (Inicia)
Propósito:	Visualizar los datos de un potencial obrero determinado.
Descripción: El caso de uso inicia cuando el actor desea visualizar los datos de un potencial obrero que se encuentran almacenados en el sistema. El caso de uso termina cuando el actor accede a otras opciones que no sean esas o cierra la aplicación.	
Referencias:	R1, R7. Este caso de uso tiene incluido el caso de uso Validar usuario.

Nombre del CUS:	Actualizar Guardia obrera
Actores del sistema:	Jefe de Seguridad y Protección. (Inicia)
Propósito:	Actualizar los datos de la guardia obrera de la universidad.

Descripción: El caso de uso inicia cuando el actor accede a la opción de insertar, eliminar, modificar la distribución de la guardia obrera. Puede insertar un nuevo servicio de guardia, modificar los datos de alguno ya existente en la Base de Datos o eliminar alguno. El caso de uso finaliza cuando el actor ha actualizado la distribución de la guardia obrera, y accede a otras opciones o cierra la aplicación.	
Referencias:	R1, R8, R8.1, R8.2, R8.3 R9. Este caso de uso tiene incluido el de Validar usuario.
Precondiciones:	Para modificar o eliminar debe existir en la Base de Datos el trabajador con el servicio de guardia y para insertar uno nuevo deben estar definidos los datos del mismo como Área, Departamento, Tipo de Servicio y Horario. El usuario debe estar autenticado como Jefe de Seguridad y Protección.
Poscondiciones:	Los datos de la guardia obrera quedaron actualizados.
Requerimientos especiales:	-

Nombre del CUS:	Visualizar guardia obrera.
Actores del sistema:	Jefe de Seguridad y Protección. (Inicia)
Propósito:	Visualizar los datos de la guardia obrera de la universidad.
Descripción: El caso de uso inicia cuando el actor desea visualizar los datos de la distribución de la guardia obrera que se encuentran almacenados en el sistema. El caso de uso termina cuando el actor accede a otras opciones que no sean esas o cierra la aplicación.	

Referencias:	R1, R10. Este caso de uso tiene incluido el caso de uso Validar usuario.
---------------------	--

Nombre del CUS:	Actualizar Área.
Actores del sistema:	Jefe de Seguridad y Protección. (Inicia)
Descripción: El caso de uso inicia cuando el actor escoge la opción de insertar, eliminar o modificar las áreas definidas para los trabajadores. Puede insertar las áreas, modificar las que haya insertado y eliminar las que ya existan en la Base de Datos. El CU finaliza cuando el actor selecciona otra opción o cierra la aplicación.	
Referencias:	R1, R11, R11.1, R11.2, R11.3. Este caso de uso tiene incluido el caso de uso Validar usuario.
Precondiciones:	Para modificar y eliminar un área debe estar ya esta en la Base de Datos. El usuario debe estar autenticado como Jefe de Seguridad y Protección.
Poscondiciones:	Las áreas quedan actualizadas.
Requerimientos especiales:	-

Nombre del CUS:	Visualizar Área.
Actores del sistema:	Jefe de Seguridad y Protección. (inicia)
Descripción: El CU inicia cuando un actor desea visualizar las áreas definidas para los trabajadores. Cuando obtiene los parámetros requeridos para la localización, consulta las tablas correspondientes y muestra el listado. El CU finaliza cuando el actor selecciona otra opción o cierra la aplicación.	

Referencias:	R1, R12. Este caso de uso tiene incluido el caso de uso Validar usuario.
---------------------	--

Nombre del CUS:	Actualizar Departamento.
Actores del sistema:	Jefe de Seguridad y Protección. (Inicia)
Descripción: El CU inicia cuando el actor escoge la opción de insertar, eliminar o modificar los departamentos definidos para los trabajadores. Puede insertar los departamentos, modificar los que haya insertado y eliminar los que ya existan en la Base de Datos. El CU finaliza cuando el actor selecciona otra opción o cierra la aplicación.	
Referencias:	R1, R13, R13.1, R13.2, R13.3. Este caso de uso tiene incluido el caso de uso Validar usuario.
Precondiciones:	Para modificar y eliminar un departamento, este debe estar en la Base de Datos. El usuario debe estar autenticado como Jefe de Seguridad y Protección.
Poscondiciones:	Los departamentos quedan actualizados.
Requerimientos especiales:	-

Nombre del CUS:	Visualizar Departamento.
Actores del sistema:	Jefe de Seguridad y Protección. (inicia)
Descripción: El CU inicia cuando un actor desea visualizar los departamentos definidos para los trabajadores. Cuando obtiene los parámetros requeridos para la localización, consulta las tablas correspondientes y muestra el listado. El CU finaliza cuando el actor selecciona otra opción o cierra la aplicación.	
Referencias:	R1, R14. Este caso de uso tiene incluido el caso de uso Validar usuario.

Nombre del CUS:	Actualizar Tipo de Servicio.
Actores del sistema:	Jefe de Seguridad y Protección. (Inicia)
Descripción: El CU inicia cuando el actor escoge la opción de insertar, eliminar o modificar los tipos de servicios de la guardia obrera definidos para los trabajadores. Puede insertar los tipos de servicios, modificar los que haya insertado y eliminar los que ya existan en la Base de Datos. El CU finaliza cuando el actor selecciona otra opción o cierra la aplicación.	
Referencias:	R1, R15, R15.1, R15.2, R15.3. Este caso de uso tiene incluido el caso de uso Validar usuario.
Precondiciones:	Para modificar y eliminar un tipo de servicio, este debe estar en la Base de Datos. El usuario debe estar autenticado como Jefe de Seguridad y Protección.
Poscondiciones:	Los tipos de servicios quedan actualizados.
Requerimientos especiales:	-

Nombre del CUS:	Visualizar Tipo de Servicio.
Actores del sistema:	Jefe de Seguridad y Protección. (inicia)
Descripción: El CU inicia cuando un actor desea visualizar los tipos de servicios definidos para los trabajadores. Cuando obtiene los parámetros requeridos para la localización, consulta las tablas correspondientes y muestra el listado. El CU finaliza cuando el actor selecciona otra opción o cierra la aplicación.	
Referencias:	R1, R16. Este caso de uso tiene incluido el caso de uso Validar usuario.

Nombre del CUS:	Actualizar Horario.
Actores del sistema:	Jefe de Seguridad y Protección. (Inicia)
Descripción: El CU inicia cuando el actor escoge la opción de insertar, eliminar o modificar los horarios de la guardia obrera definidos para los trabajadores. Puede insertar un nuevo horario, modificar los que haya insertado y eliminar los que ya existan en la Base de Datos. El CU finaliza cuando el actor selecciona otra opción o cierra la aplicación.	
Referencias:	R1, R17, R17.1, R17.2, R17.3. Este caso de uso tiene incluido el caso de uso Validar usuario.
Precondiciones:	Para modificar y eliminar un horario, este debe estar en la Base de Datos. El usuario debe estar autenticado como Jefe de Seguridad y Protección.
Poscondiciones:	Los horarios quedan actualizados.
Requerimientos especiales:	-

Nombre del CUS:	Visualizar Horario.
Actores del sistema:	Jefe de Seguridad y Protección. (inicia)
Descripción: El CU inicia cuando un actor desea visualizar los horarios definidos para los trabajadores. Cuando obtiene los parámetros requeridos para la localización, consulta las tablas correspondientes y muestra el listado. El CU finaliza cuando el actor selecciona otra opción o cierra la aplicación.	
Referencias:	R1, R18. Este caso de uso tiene incluido el caso de uso Validar usuario.

Nombre del CUS:	Actualizar Ubicación.
Actores del sistema:	Puesto de Mando. (Inicia)
Descripción: El CU inicia cuando el actor escoge la opción de insertar, eliminar o modificar las ubicaciones de los trabajadores para la realización de su guardia correspondiente. Puede insertar una nueva ubicación, modificar la que haya insertado y eliminar las que ya existan en la Base de Datos. El CU finaliza cuando el actor selecciona otra opción o cierra la aplicación.	
Referencias:	R1, R19, R19.1, R19.2, R19.3. Este caso de uso tiene incluido el caso de uso Validar usuario.
Precondiciones:	Para modificar y eliminar una ubicación, este debe estar en la Base de Datos. El usuario debe estar autenticado como Puesto de Mando.
Poscondiciones:	Las ubicaciones quedan actualizadas.
Requerimientos especiales:	-

Nombre del CUS:	Visualizar Ubicación.
Actores del sistema:	Puesto de Mando. (inicia)
Descripción: El CU inicia cuando un actor desea visualizar las ubicaciones de los trabajadores para la realización de su guardia correspondiente. Cuando obtiene los parámetros requeridos para la localización, consulta las tablas correspondientes y muestra el listado. El CU finaliza cuando el actor selecciona otra opción o cierra la aplicación.	
Referencias:	R1, R20. Este caso de uso tiene incluido el caso de uso Validar usuario.

Nombre del CUS:	Actualizar guardia estudiantil
------------------------	--------------------------------

Actores del sistema:	Coordinador de Año. (Inicia)
Propósito:	Actualizar los datos de la guardia estudiantil de la facultad.
Descripción: El CU inicia cuando el actor accede a la opción de insertar, eliminar, modificar la distribución de la guardia estudiantil. Puede insertar un nuevo estudiante, modificar los datos de alguno ya existente en la Base de Datos o eliminar alguno. El CU finaliza cuando el actor ha actualizado la distribución de la guardia estudiantil, y accede a otras opciones o cierra la aplicación.	
Referencias:	R1, R21, R21.1, R21.2, R21.3, R22. Este caso de uso tiene incluido el de Validar usuario.
Precondiciones:	Para modificar o eliminar debe existir en la Base de Datos el estudiante y para insertar uno nuevo deben estar definidos los datos del mismo como Ubicación y Horario. El usuario debe estar autenticado como Coordinador de Año.
Poscondiciones:	Los datos de la guardia estudiantil quedaron actualizados.
Requerimientos especiales:	-

Nombre del CUS:	Visualizar guardia estudiantil.
Actores del sistema:	Coordinador de Año. (Inicia)
Propósito:	Visualizar los datos de la guardia estudiantil de la facultad.

Descripción: El CU inicia cuando el actor desea visualizar los datos de la distribución de la guardia estudiantil de la facultad que se encuentran almacenados en el sistema. El caso de uso termina cuando el actor accede a otras opciones que no sean esas o cierra la aplicación.	
Referencias:	R1, R23. Este caso de uso tiene incluido el caso de uso Validar usuario.

Nombre del CUS:	Insertar cumplimiento guardia obrera y estudiantil
Actores del sistema:	Puesto de Mando. (Inicia). Jefe de Seguridad y Protección
Propósito:	Insertar el cumplimiento de la guardia obrera y estudiantil.
Descripción: El CU inicia cuando el actor Puesto de Mando accede a la opción de insertar cumplimiento de la guardia obrera y estudiantil. El CU finaliza cuando el actor ha insertado o no el cumplimiento, y accede a otras opciones o cierra la aplicación.	
Referencias:	R1, R24. Este caso de uso tiene incluido el de Validar usuario.

Precondiciones:	Para insertar debe existir en la Base de Datos el trabajador o estudiante y deben estar definidos los datos del mismo como Área, Departamento, Tipo de Servicio, Ubicación y Horario. El usuario debe estar autenticado como Puesto de Mando. El usuario debe estar autenticado como Puesto de Mando o Jefe de Seguridad y Protección.
Poscondiciones:	El cumplimiento de la guardia obrera y estudiantil queda insertado.
Requerimientos especiales:	-

Nombre del CUS:	Visualizar cumplimiento guardia obrera y estudiantil
Actores del sistema:	Puesto de Mando. (Inicia). Jefe de Seguridad y Protección
Propósito:	Visualizar el cumplimiento de la guardia obrera y estudiantil.
Descripción: El CU inicia cuando el actor Puesto de Mando desea visualizar el cumplimiento de la guardia obrera y estudiantil que se encuentra almacenado en el sistema. El caso de uso termina cuando el actor accede a otras opciones que no sean esas o cierra la aplicación.	

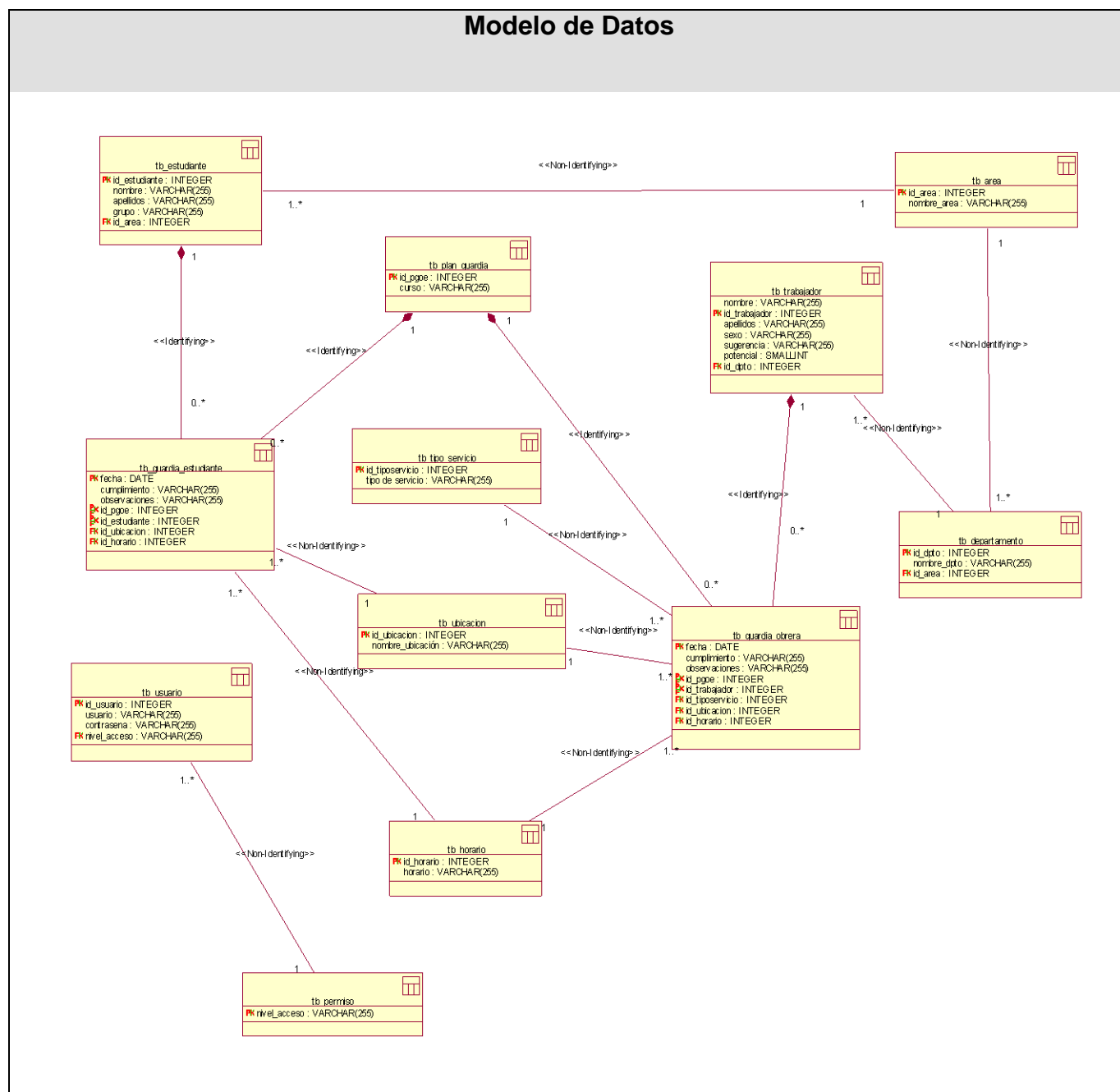
Referencias:	R1, R25. Este caso de uso tiene incluido el caso de uso Validar usuario.
---------------------	--

Nombre del CUS:	Listar historial de servicios de guardia por diferentes criterios
Actores del sistema:	Jefe de Seguridad y Protección. (inicia)
Descripción: El CU inicia cuando un actor desea listar historial de servicios de guardia por diferentes criterios. Cuando obtiene los parámetros requeridos para la localización, consulta las tablas correspondientes y muestra el listado. El CU finaliza cuando el actor selecciona otra opción o cierra la aplicación.	
Referencias:	R1, R26. Este caso de uso tiene incluido el caso de uso Validar usuario.

Nombre del CUS:	Mostrar reporte guardia obrera y estudiantil
Actores del sistema:	Jefe de Seguridad y Protección. (Inicia).
Propósito:	Mostrar reportes de la guardia obrera y estudiantil

Descripción: El CU inicia cuando el actor accede a la opción de mostrar reporte del registro de la guardia obrera y estudiantil. El CU finaliza cuando se ha obtenido el documento con el reporte de la guardia obrera y estudiantil, y el actor accede a otras opciones o cierra la aplicación.	
Referencias:	R1, R27. Este caso de uso tiene incluido el de Validar usuario.
Precondiciones:	Para mostrar reporte deben existir los datos Nombre, Apellidos, Grupo. Fecha, Horario, Ubicación y Tipo de Servicio. El usuario debe estar autenticado como Jefe de Seguridad y Protección.
Poscondiciones:	Se muestra el documento con el reporte de la guardia obrera y estudiantil.
Requerimientos especiales:	-

Anexo 3. Modelo de Datos



Anexo 4. Encuesta para la determinación de expertos

Encuesta para determinar el coeficiente de competencia del experto.

Nombre y apellidos: _____

Usted ha sido seleccionado como posible experto para ser consultado respecto al grado de relevancia del Sistema para la gestión de información de la guardia obrera y estudiantil que presumiblemente debe servir para ayudar a la toma de decisiones en el proceso de gestión de esta información.

Necesitamos antes de realizarle la consulta correspondiente como parte del método empírico de investigación “consulta a expertos”, determinar su coeficiente de competencia en este tema, a los efectos de reforzar la validez del resultado de la consulta que realizaremos. Por esta razón le rogamos que responda las siguientes preguntas de la forma más objetiva que le sea posible.

1.- Marque con una cruz (X), en la tabla siguiente, el valor que se corresponde con el grado de conocimientos que usted posee sobre el tema “Sistemas de Gestión de Información”. Considere que la escala que le presentamos es ascendente, es decir, el conocimiento sobre el tema referido va creciendo desde 0 hasta 10.

1	2	3	4	5	6	7	8	9	10

2.- Realice una autovaloración del grado de influencia que cada una de las fuentes que le presentamos a continuación, ha tenido en su conocimiento y criterio sobre el tema “Sistemas de Gestión de Información”. Para ello marque con una cruz (X), según corresponda, en A (alto), M (medio) o B (bajo).

Fuentes de argumentación.	Grado de influencia de cada una de las fuentes.		
	A (alto)	M (medio)	B (bajo)
Análisis teórico realizado por usted.			

Su experiencia obtenida.			
Trabajo de autores nacionales.			
Trabajo de autores extranjeros.			
Su propio conocimiento del estado del problema en el extranjero.			
Su intuición.			

Muchas gracias.

Anexo 5. Encuesta a expertos

Como parte de un tema de tesis se desarrolló un **Sistema para favorecer la gestión de información de la guardia obrera y estudiantil en la UHo**, el mismo está siendo sometido a su consideración para obtener el grado de relevancia de su desarrollo, así la información que brinde es esencial con este fin.

A continuación se presentan las categorías de evaluación y las tablas a ser evaluadas:

MR: Muy relevante. **BR:** Bastante relevante. **R:** Relevante. **PR:** Poco relevante **NR:** No relevante.

Marque con una X la categoría que considera adecuada tecnológicamente para garantizar el funcionamiento del sistema:

Criterios tecnológicos	MR	BR	R	PR	NR
Intuitividad de interface Web					
Metodología RUP + UML					
Framework Ajax ZK					
BBDD PostgreSQL					
Validación por criterio de expertos					
Encuestas y entrevistas (requisitos)					
Valoración de sostenibilidad					
Valoración de satisfacción - usuarios					
Reportes en JasperReports					
IDEs Eclipse e iReport					
Presentación de interface Web					
Usabilidad					
Criterios funcionales					
Actualizar potencial obrero					
Actualizar guardia estudiantil					
Actualizar guardia obrera					

Visualizar cumplimiento guardia obrera					
Visualizar cumplimiento guardia estudiantil					
Realizar búsqueda por criterios					
Mostrar reportes					

Escriba a continuación qué aspectos considera que deben ser incluidos o eliminados en esta propuesta:

Aspectos que se proponen ser incluidos	Aspectos que se proponen ser eliminados

Señale si considera que el nombre de alguno de los aspectos de la propuesta, debe cambiarse

Aparece como	Debe ser cambiado por

Otra sugerencia que usted desee hacer sobre la propuesta el Sistema para la gestión de información de la guardia obrera y estudiantil que estamos sometiendo a su consideración:

Muchas gracias

Anexo 6. Encuesta a usuarios

Nombre y apellidos: _____.

Institución a la que pertenece: _____.

Cargo actual: _____.

Calificación profesional, grado científico o académico: _____.

Profesor: _____.

Licenciado: _____.

Especialista: _____.

Máster: _____.

Doctor: _____.

Años de experiencia en el cargo: _____.

Años de experiencia docente y/o en la investigación: _____.

Como parte del tema de tesis en la Universidad de Holguín se elaboró un Sistema para la gestión de información de la guardia obrera y estudiantil con el objetivo de mejorar estos procesos. Se anexa a esta encuesta pantallas del sistema en caso de que desee consultarlas. Se requiere su opinión con relación a:

- Grado de relevancia de los aspectos de los módulos del sistema en cuanto al apoyo a la gestión de la información de la guardia obrera y estudiantil.
- ¿Qué aspectos cree Ud. que deben ser incluidos o eliminados?
- Sugerencias de cambios de denominación de los aspectos propuestos, cuyo grado de relevancia, sometemos a su consideración.

Indicaciones:

A continuación le presentamos una tabla que contiene los aspectos de los módulos del Sistema de gestión de información de la guardia obrera y estudiantil. A la derecha aparece la escala:

MR: Muy relevante. **BR:** Bastante relevante. **R:** Relevante. **PR:** Poco relevante
NR: No relevante.

Marque con una cruz (X) en la celda que se corresponda con el grado de relevancia que usted otorga a cada aspecto del sistema.

Le agradecemos anticipadamente el esfuerzo que sabemos hará para responder, con la mayor fidelidad posible a su manera de pensar la presente encuesta.

Muchas gracias.

Guardia obrera y estudiantil	MR	BR	R	PR	NR
Visualizar Potencial Obrero					
Visualizar Guardia Obrera					
Visualizar Guardia Estudiantil					
Insertar cumplimiento guardia obrera y estudiantil					
Visualizar cumplimiento guardia obrera y estudiantil					
Visualizar listado de servicios de guardia por diferentes criterios					

Escriba a continuación qué aspectos considera que deben ser incluidos o eliminados en esta propuesta:

Aspectos que se proponen ser incluidos	Aspectos que se proponen ser eliminados

Señale a continuación, si considera que el nombre de alguno de los aspectos de la propuesta, debe ser cambiado:

El momento aparece como:	El momento debe ser cambiado por:

Otra sugerencia que usted desee hacer sobre la propuesta el Sistema de gestión de información de la guardia obrera y estudiantil que estamos sometiendo a su consideración:
