

Universidad de las Ciencias Informáticas

Facultad 3



Título: “Desarrollo del módulo GOE para el Sistema de Administración y Economía de la Facultad 3”

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

Autores:

Oneisy Aguilar López

Luis Angel Castillo Rodriguez

Tutores:

Ing. Arnolis Salgueiro Arzuaga

Ing. Yunet Suárez Abrante

La Habana, Junio de 2015



“Déjenme decirles, a riesgo de parecer ridículo, que el revolucionario verdadero está guiado por sentimientos de amor. Sean capaces siempre de sentir, en lo más hondo, cualquier injusticia realizada contra cualquiera, en cualquier parte del mundo. Es la cualidad más linda del revolucionario”

Ernesto Che Guevara

AGRADECIMIENTOS

Agradecer primeramente ante todo a mi familia, principalmente a mi mamá, mi papá, mi hermano, mis tíos, mi abuela, mis primos por ser ellos, el motor impulsor de mi vida, guiándome y aconsejándome en cada paso dado y sobre todo a mi mamá y mi papá por haber dejado sus vidas a un lado para dedicarla a mí. A mis viejos amigos que ya no se encuentran aquí, por haber formado parte de mi vida y estar conmigo cuando lo necesité. A los nuevos amigos de la brigada 3507: mis compañeros de aula, por haber formado parte de mis nuevas aventuras y haberme ayudado cuando lo necesité. A todas mis amistades de Mataguá y Seibabo, que por la lejanía no pudieron compartir este momento conmigo, pero siempre estuvieron presente. A mis compañeras de cuarto por soportarme con tanto cariño. A mi pareja, por soportar mis altas y bajas y estar conmigo por más de 4 años y seguirme queriendo y a su familia por haberme aceptado. A Luis Angel y a Ivaniyet por haberme ayudado a cruzar el camino de la programación, porque sin ellos no hubiese podido seguir adelante. A Ivaniyet por ser mi hermano en la universidad y haberme enseñado todo lo que se hoy. A Darianis por soportarme y haber sido mi mejor amiga, un poco peleona, pero gracias por tus consejos. A Yudi y Raquel por haber sido mis hermanas de la universidad y estar conmigo en las buenas y en las malas. Agradezco a todos los presentes por estar aquí y confiar en nosotras. A todos los que no mencioné para que no se sientan olvidados. A la universidad por haberme dado la oportunidad de seguir superándome en la vida.

Dedicatoria

Dedico este trabajo a mi mamá, mi papá, mi hermano, mis tíos, mi abuela, mis primos, por desear y hacer cumplir mi sueño de convertirme en una ingeniera, besos, los amo mucho.

Oneisy

AGRADECIMIENTOS

Agradecer primeramente ante todo a mi mamá, mi papá y mis hermanos por ser ellos la guía de mi vida. A los compañeros de la brigada 3507: por haber formado parte de mis aventuras. A todas mis amistades que no pudieron compartir este momento conmigo y los que comparten conmigo hoy. A mi pareja, por soportarme y estar conmigo por más de 4 años y seguirme queriendo.

Dedicatoria

Dedico este trabajo a mi mamá, mi papá, mis hermanos, los amo mucho.

Luis Angel

DECLARACIÓN JURADA DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Oneisy Aguilar López
Autora

Luis Angel Castillo Rodriguez
Autor

Ing. Arnolis Salgueiro Arzuaga
Tutor

Ing. Yunet Suárez Abrante
Tutora

DATOS DE CONTACTO

Tutora: Ing. Yunet Suárez Abrante

Correo electrónico: yabrante@uci.cu

Tutor: Ing. Arnolis Salgueiro Arzuaga

Correo electrónico: asalgueiro@uci.cu

Autora: Oneisy Aguilar López

Correo electrónico: oalopez@estudiantes.uci.cu

Autor: Luis Angel Castillo Rodriguez

Correo electrónico: lacastillo@estudiantes.uci.cu

RESUMEN

El presente trabajo surge a partir de la necesidad de mejorar la disponibilidad y control de la información de los procesos asociados a la gestión de la Guardia Obrera Estudiantil y la integración de varios módulos en el nuevo Sistema de Administración y Economía de la Facultad 3 de la Universidad de las Ciencias Informáticas (UCI). La investigación tiene como objetivo desarrollar una aplicación web para la gestión de los procesos de la Guardia Obrera Estudiantil en la Facultad 3, de forma tal que contribuya al adecuado funcionamiento y organización para la accesibilidad y control de la información de dichos procesos.

Como resultado final de la investigación se obtiene el módulo GOE para el nuevo Sistema de Administración y Economía, capaz de centralizar y gestionar la información de los procesos asociados a la Guardia Obrera Estudiantil adscritos a la Facultad 3 de la Universidad de las Ciencias Informáticas.

Palabras claves: control, guardia obrera estudiantil, aplicación web, planificación.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción	5
1.2 Marco teórico. Conceptos y definiciones	5
1.3 Análisis de sistemas informáticos de gestión de guardia	6
1.3.1 Sistemas informáticos de gestión de guardia internacionales	6
1.3.2 Sistemas informáticos de gestión de guardia nacionales	7
1.4 Metodología de desarrollo de software	11
1.5 Tecnologías y herramientas de desarrollo	13
1.6 Conclusiones del capítulo	17
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN	18
2.1 Introducción	18
2.2 Propuesta de solución	18
2.3 Fase de Ejecución de la metodología seleccionada	18
2.3.1 Modelado del negocio	18
2.3.2 Ingeniería de Requisitos	24
2.3.3 Análisis y diseño	27
2.3.4 Modelo de implementación	40
2.4 Conclusiones del capítulo	45
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN	46
3.1 Introducción	46
3.2 Estrategia de pruebas	46
3.3 Niveles de prueba	46
3.4 Métodos de prueba	48
3.4.1 Pruebas de caja blanca o estructural	48
3.4.2 Pruebas de caja negra o funcional	49
3.4.3 Pruebas de aceptación	49
3.5 Pruebas realizadas al sistema	51
3.6 Conclusiones del capítulo	56
CONCLUSIONES	57
RECOMENDACIONES	58

REFERENCIAS BIBLIOGRÁFICAS	59
ANEXOS	62

ÍNDICE DE FIGURAS

Figura 1: Modelo Conceptual.....	19
Figura 2: Diagrama de proceso: Planificar GOE.....	21
Figura 3: Funcionamiento general del patrón MVC.	29
Figura 4: Diagrama de clases del Diseño: Gestionar planificación de la GOE.	31
Figura 5: Patrón Experto.	33
Figura 6: Patrón Bajo Acoplamiento y Alta Cohesión.	33
Figura 7: Patrón Creador.....	33
Figura 8: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.	35
Figura 9: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación.	36
Figura 10: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.	36
Figura 11: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización.	38
Figura 12: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de mantenimiento.	38
Figura 13: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.	38
Figura 14: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas.	39
Figura 15: Modelo de datos.....	40
Figura 16: Diagrama de componentes.....	44
Figura 17: Diagrama de despliegue.....	45
Figura 18: Niveles de Prueba de Software.	47
Figura 19: Métodos de prueba caja blanca y negra.	48
Figura 20: Grafo formado mediante la técnica del camino básico.	49
Figura 21: Segmento de código seleccionado.	51
Figura 22: Grafo formado luego de aplicar la técnica del camino básico al código seleccionado.	52
Figura 23: Condiciones para que se pueda ejecutar la guardia.....	53
Figura 24: Escenarios del caso de prueba Planificar GOE.....	53

ÍNDICE DE TABLAS

Tabla 1: Comparación entre los sistemas informáticos.	10
Tabla 2: Comparación entre las metodologías.	11
Tabla 3: Descripción de las fases AUP variación para la UCI.	12
Tabla 4: Descripción del proceso de negocio: Planificar GOE.	23
Tabla 5: Requisitos No Funcionales.	26
Tabla 6: Historia de Usuario: Planificar GOE.	27
Tabla 7: Tamaño Operacional de Clase (TOC).	35
Tabla 8: Rango de valores para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de implementación y Reutilización) relacionados con la métrica TOC.	35
Tabla 9: Relaciones entre clases.	37
Tabla 10: Rango de valores para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas) relacionados con la métrica RC.	37
Tabla 11: Cuadro lógico de Iadov.	50
Tabla 12: Índice de satisfacción grupal.	50
Tabla 13: Iteraciones de pruebas funcionales.	54
Tabla 14: Cuadro lógico de Iadov completado.	55
Tabla 15: Resultado de aplicación de la técnica Iadov.	55

INTRODUCCIÓN

Las últimas décadas del siglo XX y los primeros años del siglo XXI han estado marcados por un evidente cambio en el carácter y complejidad de las amenazas que ponen en riesgo la seguridad de las organizaciones y de la sociedad. Este es uno de los temas principales a los que las entidades le dedican gran parte de sus esfuerzos y recursos por la importancia que tiene para la preservación de su entorno organizacional.

En Cuba, las actividades delictivas en el sector estatal ocasionan cuantiosas pérdidas económicas y contribuyen al desvío de recursos, indisciplinas sociales y pérdida de valores. Con el objetivo de enfrentar estas amenazas económicas y sociales, cada institución del estado desarrolla un plan de Seguridad y Protección para las personas, locales y recursos disponibles, que incluye trabajadores, directivos, estudiantes, la sociedad en general.

La Universidad de las Ciencias Informáticas (UCI) no está exenta de estos temas de indisciplinas que se alejan de la ética y atentan directamente contra la economía del país. Es por ello, que uno de los métodos empleados para proteger los recursos que pertenecen a la Universidad es la Guardia Obrera Estudiantil (GOE).

La GOE es parte integrante del Sistema de Seguridad y Protección y constituye un elemento importante en la protección interna de la UCI. Su objetivo es contribuir a la protección y cuidado de las personas, recursos y medios materiales de la Universidad, dando respuesta ante cualquier hecho delictivo que suceda. (1)

El Vicedecanato de Economía y Administración dirige y controla los procesos asociados a la Guardia Obrera Estudiantil de la Facultad 3, entre los cuales se encuentran: planificación, control del cumplimiento y ejecución de la misma. En cursos anteriores se desarrollaron sistemas para la gestión de estos procesos a nivel de universidad y facultad. Según el documento Informe de Administración Guardia Obrera Estudiantil, emitido en el curso 2013- 2014 por el vicedecanato, estos sistemas no cumplen con las necesidades, especificidades ni con la flexibilidad que deben brindar a los procesos de planificación, ejecución y control hoy día, volviéndose inutilizables y deficientes. Algunas de las deficiencias encontradas son:

1. Problemas en la planificación de la Guardia Obrera Estudiantil que no reflejan concesiones establecidas respecto a: rotaciones por turnos, parejas de guardias, repetición de turnos, días y postas. (1)
2. Existencia de duplicaciones de notificaciones a los usuarios sobre el día planificado para la Guardia Obrera Estudiantil que no se corresponden, en cada caso, generando incertidumbre. (1)

3. Los sistemas no conciben el principio contraído con los trabajadores de parejas de guardia por afinidad. (1)
4. Planificación de Guardia Obrera Estudiantil a personal no disponible por falta de actualización de los recursos humanos para el desarrollo de la misma. (1)

Actualmente los procesos de planificación y control se realizan de forma manual, por lo que el acceso a la información es difícil, existe desactualización y pérdida de datos. Esto conlleva a que no se envíe la planificación con los 3 meses de antelación establecidos y que se dificulte la obtención de reportes de información necesarios para el control de la misma. Se desea por otro lado que el módulo GOE forme parte del Sistema de Administración y Economía de la Facultad 3, arrojando la necesidad de desarrollar un nuevo módulo que responda a la problemática planteada.

Lo antes expuesto constituye la problemática que origina esta investigación de la cual se define el siguiente **problema a resolver**: ¿Cómo gestionar la información asociada a los procesos de planificación, ejecución y control de la Guardia Obrera Estudiantil de la Facultad 3, de manera que mejore la disponibilidad y control de la información?

Según el problema identificado, el **objeto de estudio** es: procesos de planificación, ejecución y control asociados a la gestión de Guardia Obrera Estudiantil.

Mientras que el **campo de acción** está enmarcado en los procesos de planificación, ejecución y control asociados a la gestión de la Guardia Obrera Estudiantil de la Facultad 3.

En correspondencia con el problema a resolver se plantea como **objetivo general**: Desarrollar el módulo GOE para el Sistema de Administración y Economía de la Facultad 3, de manera que mejore la disponibilidad y control de la información. Para darle cumplimiento al objetivo general propuesto se han definido los siguientes **objetivos específicos**:

1. Definir el marco teórico de la investigación mediante el estudio y el análisis de los principales referentes teóricos para el desarrollo de la solución.
2. Realizar el análisis y diseño de los procesos de planificación, ejecución y control asociados a la gestión de la Guardia Obrera Estudiantil en la Facultad 3.
3. Implementar la solución para obtener los componentes de software de los procesos de planificación, ejecución y control asociados a la gestión de la Guardia Obrera Estudiantil en la Facultad 3.
4. Validar la solución propuesta mediante la realización de pruebas de caja blanca y caja negra.

Para darle cumplimiento a los objetivos específicos se plantean las **tareas de la investigación**:

1. Análisis de sistemas informáticos existentes a nivel nacional e internacional.

2. Caracterización de la metodología, las tecnologías y las herramientas a utilizar para el desarrollo de la solución.
3. Análisis del negocio de los procesos: planificación, control del cumplimiento y ejecución de la GOE.
4. Levantamiento de requisitos de los procesos: planificación, control del cumplimiento y ejecución de la Guardia Obrera Estudiantil.
5. Diseño de la estructura y comportamiento de los componentes de los procesos: planificación, control del cumplimiento y ejecución de la Guardia Obrera Estudiantil.
6. Validación del diseño propuesto.
7. Diseño del modelo de datos.
8. Diseño del diagrama de componentes y despliegue del sistema.
9. Implementación de los componentes diseñados.
10. Validación de la implementación realizada.

Se toma como **idea a defender** que la implementación del módulo GOE para el Sistema de Administración y Economía de la Facultad 3 contribuirá a mejorar la disponibilidad y control de la información asociada a los procesos de planificación, ejecución y control de la Guardia Obrera Estudiantil.

Posibles resultados: Módulo GOE para el Sistema de Administración y Economía de la Facultad 3 (SAEF3).

Métodos Científicos

Métodos teóricos:

Análisis histórico-lógico: permitirá al equipo de desarrollo realizar un estudio de las aplicaciones existentes a nivel nacional e internacional, especialmente en la UCI, para analizar las irregularidades, el comportamiento y la planificación de los procesos de la Guardia Obrera Estudiantil.

Modelación: se reflejará mediante el análisis y diseño del módulo GOE de la Facultad 3 a través de las diferentes tecnologías y herramientas que se seleccionarán, así como de la implementación de la aplicación.

Analítico-sintético: facilitará el análisis de aspectos generales relacionados con los principales conceptos a investigar, lo que contribuirá a la unificación de información sobre el tema de la Guardia Obrera Estudiantil y a la selección de los aspectos esenciales para la elaboración del primer capítulo.

Métodos Empíricos:

Entrevista: se aplicó a la Vicedecana de Economía y Administración donde se identificaron las deficiencias que dificultan el proceso de gestión de la Guardia Obrera Estudiantil de la Facultad 3 y la información de las necesidades existentes en el desarrollo del módulo y los requerimientos tanto funcionales como no funcionales que debe cumplir la aplicación.

El trabajo de diploma se encuentra estructurado en tres capítulos de la siguiente manera:

Capítulo 1: Fundamentación Teórica. Se definen los principales conceptos que se relacionan con la investigación para lograr un mejor entendimiento del negocio, lo que incluye el análisis de sistemas existentes a nivel nacional e internacional para la gestión de la Guardia Obrera Estudiantil. Se realiza una descripción de la metodología escogida por el equipo de desarrollo en conjunto con el cliente para su utilización en el desarrollo de la solución, así como la descripción de las herramientas y tecnologías de desarrollo escogidas.

Capítulo 2: Propuesta de Solución. En este capítulo se describen elementos como: la propuesta de solución de la aplicación a desarrollar, el modelo conceptual que refleja las entidades que están presentes en el negocio y cómo se relacionan, los procesos de negocio, técnicas de captura y validación de requisitos, requisitos funcionales y no funcionales, diseño arquitectónico a emplear en el desarrollo de la solución, diagrama de clases del diseño según el marco de trabajo que se utiliza y patrones de diseño. Se realiza la validación del diseño mediante el uso de diferentes métricas y se describe el modelo de datos. Por otra parte, se elabora el modelo de implementación el cual está representado por el diagrama de componentes y despliegue para una mayor visualización de la interacción con el sistema y el correcto funcionamiento de sus nodos. Además se describen los estándares de codificación para lograr una eficiente implementación de la aplicación.

Capítulo 3: Validación de los Resultados. En este capítulo se muestran los resultados de las pruebas realizadas al sistema informático obtenido mediante las estrategias de prueba. Las mismas se dividen en niveles de pruebas y cada nivel tiene un método de prueba. De estos métodos se explica su funcionamiento y resultados a través de las pruebas realizadas al software para detectar errores relacionados con sus funcionalidades y para comprobar la veracidad del código.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se definen los principales conceptos que se relacionan con la investigación, incluye el análisis de sistemas existentes a nivel nacional e internacional para la gestión de la Guardia Obrera Estudiantil. Se realiza una descripción de la metodología escogida por el equipo de desarrollo en conjunto con el cliente para su utilización en el desarrollo de la solución, así como la descripción de las herramientas y tecnologías de desarrollo escogidas.

1.2 Marco teórico. Conceptos y definiciones

El análisis conceptual tiene gran connotación en los trabajos investigativos y evita la confusión de términos entre diferentes conceptos. A continuación, se describen los principales conceptos que a criterio del equipo de desarrollo son los que mayor aporte brindan a la investigación.

Proceso: es una competencia que tiene la organización que representa un conjunto de actividades, interacciones y recursos con una finalidad común: transformar las entradas en salidas que agreguen valor a los clientes. El proceso es realizado por personas organizadas según una cierta estructura, tienen tecnología de apoyo y manejan información. (2)

Gestión de procesos: es una disciplina de gestión que ayuda a la dirección de la empresa a identificar, representar, diseñar, formalizar, controlar, mejorar y hacer más productivos los procesos de la organización para lograr la confianza del cliente. (2)

Información: es un dato o un conjunto de datos evaluados por un individuo concreto, en un momento dado, sobre un problema específico, para alcanzar un objetivo determinado. Es un recurso crítico de la organización, pues los restantes recursos no pueden ser administrados sin ella. La información se genera a partir de un grupo de datos seleccionados para reducir la dosis de ignorancia o el grado de incertidumbre de quien debe adoptar una decisión. Para que la información resulte eficiente, debe reunir una serie de requisitos, de modo tal que la utilidad que proporcione justifique el empleo de los recursos que se hubieran aplicado para producirla. (3).

Gestión de la información: constituye un proceso mediante el cual se planifican, organizan, dirigen y controlan los recursos de información de una organización asegurando un adecuado tratamiento, intercambio y uso de este recurso, que contribuyan al establecimiento de fortalezas organizacionales y permitiendo que todos sus miembros dispongan de forma adecuada de los recursos informativos que existen en ella. (4)

Planificación: es el establecimiento de objetivos, metas, una actividad continua y unitaria que no termina con la formulación de un plan determinado, sino que implica un reajuste permanente entre medios, actividades, fines, caminos, procedimientos y la elección de los medios más convenientes para alcanzarlos como elementos comunes e importantes a la hora de definir la planificación. (5)

Control: es el conjunto de actividades que se emprenden, para medir, examinar y evaluar la ejecución de los planes y los resultados obtenidos, con el fin de detectar y prever desviaciones, diagnosticando la razón de las desviaciones y tomando las medidas correctivas necesarias para asegurar la obtención de los objetivos. (6) (7)

1.3 Análisis de sistemas informáticos de gestión de guardia

A continuación, se hace un estudio de sistemas informáticos existentes en el mundo y en Cuba.

1.3.1 Sistemas informáticos de gestión de guardia internacionales

Sistema Web de Asignación Automática de Mallas de Turnos para Protector Security (AutoMaT P.S)

Sistema web online que asigna automáticamente los turnos para Protector Security, una empresa que presta servicios de seguridad y vigilancia en diversas ciudades de Chile, maneja distintos niveles de usuarios, genera reportes mensuales, etc. De esta forma, los supervisores no tienen la facultad de asignar los horarios de trabajo, evitando pérdidas económicas por turnos inexistentes, generación de errores, ineficiencia y pérdida de recursos. Este producto tiene soporte web, está disponible 7 días x 24 horas en condiciones normales y contempla un uso óptimo de las conexiones a la base de datos. Es una aplicación que requiere de buenas prestaciones a internet así como requerimiento de hardware máximas aunque presenta una escasa documentación técnica de la misma. (8)

Gestión de calendarios de personal SAP HCM¹ en ERP 6.0 (GCP)

Solución web como cliente ligero de uso distribuido desarrollada en la plataforma Microsoft .NET e integrado online con el módulo de Gestión de Tiempos (PT) de SAP (Sistemas, Aplicaciones y Productos) HCM (Administración de Capital Humano) en ERP 6.0. Desarrollada para optimizar los procesos de planificación y gestión de calendarios de todos sus empleados sobre su sistema de gestión corporativo: SAP HCM en ERP 6.0. Puso a disposición de los distintos usuarios, de variado perfil y necesidades, una herramienta que permite la planificación de tiempos de sus equipos de trabajo, introduciendo cambios de turnos, guardias, absentismos y otras incidencias. Es una

¹ **SAP:** Systeme, Anwendungen und Produkte (Sistemas, Aplicaciones y Productos), es el primer proveedor de aplicaciones de software empresarial en el mundo. **HCM** por sus siglas en inglés Human Capital Management significa Administración de Capital Humano. La solución SAP ERP Human Capital Management ofrece un completo e integrado conjunto de herramientas para ayudarle a realizar una gestión efectiva de sus empleados.

aplicación que requiere de buenas prestaciones en la red y por tanto requerimiento de hardware máximas. (9)

TecnoHospital

Coprava (Compañía para la Organización y Programación Avanzada) ha presentado un revolucionario sistema de gestión web automático que permite planificar los turnos de personal en cualquier empresa y que permite reducir un 70% el tiempo dedicado a esta tarea. Esta aplicación web genera automáticamente el portal de trabajo para su integración en la Intranet de la compañía y el portal de consulta para los empleados. Apuesta así por el ahorro de tiempo, la facilidad de trabajo, la reducción de errores y, por supuesto, la movilidad como principales armas de este sistema. Gestiona las guardias. La aplicación es 100% online, por lo que permite a los responsables de personal manejar la información desde cualquier parte del mundo y a los trabajadores consultarla. Genera automáticamente la planificación de turnos por defecto en base a unos parámetros definidos inicialmente y que pueden ser modificados con posterioridad para conseguir la planificación deseada. Exporta informes detallados por turno, por trabajador, por área o centro, de tal forma que es posible manejar la información en diferentes formatos para evitar errores. (10)

1.3.2 Sistemas informáticos de gestión de guardia nacionales

Gestor Web para el Control de la Guardia Obrera de la UCI (CGO-UCI)

Aplicación desarrollada en la UCI con el objetivo de mejorar la obtención de reportes referente al proceso de control de la Guardia Obrera de la universidad. Es independiente del sistema operativo donde se ejecute y presenta un requerimiento de hardware mínimo, brinda escasa información sobre todo el tema perteneciente al control de la guardia obrera, disminuye el trabajo manual y agiliza el flujo de información. Esta aplicación se encarga de automatizar el trabajo de las entidades relacionadas con el proceso de control de la guardia obrera así como la generación de reportes con los resultados de este proceso. Ofrece la posibilidad de realizar reportes semanalmente relacionados con todo el proceso de control de la guardia obrera haciendo que su generación resulte menos compleja. (11).

Aplicación Web para el control de la Guardia Estudiantil y la Cuartelería de la Facultad 7 (CGEC-Fac7)

Es una aplicación web que visualiza la gestión de la información y planificación de la guardia estudiantil, la misma persigue el objetivo de proporcionarle a la dirección de la facultad la posibilidad de tener registrada toda la información necesaria referente a los procesos de la guardia estudiantil. Lo que posibilitará realizar de forma más eficiente el manejo y gestión de la información de estas actividades, asimismo proporcionará que se pueda llevar un control personalizado de la evaluación de

los estudiantes en la residencia estudiantil, generar un conjunto de reportes en formato digital acerca de estas actividades que pueden ser impresos, facilitando la planificación y el control de la Guardia Estudiantil a los responsables de estas tareas. Es independiente del sistema operativo donde se ejecute y presenta un requerimiento de hardware mínimo (12).

Sistema de gestión docente metodológica integrado con la Plataforma de Servicios Integrados para la Facultad 3 (SisGes-DM)

Aplicación informática que dio solución a las diferentes deficiencias concerniente a los procesos Plan de trabajo metodológico, Guardia docente y Control a clases, permitiendo la centralización, así como una adecuada organización y una eficiente gestión de la información. Además provee una retroalimentación de datos entre los involucrados, la generación de estadísticas, una mayor disponibilidad y actualización de la información. Es independiente del sistema operativo donde se ejecute y presenta un requerimiento de hardware mínimo. Se obtuvo una aplicación web capaz de centralizar y gestionar la información de los procesos docentes metodológicos y de Guardia docente adscritos a la Facultad 3 de la UCI. (13)

Aplicación web para la gestión de la planificación de la guardia obrera estudiantil en la UCI (PGOE-UCI)

Es un sistema informático desarrollado para la gestión de la planificación de la Guardia Obrera Estudiantil en la Universidad de las Ciencias Informáticas. De esta forma se brinda una posible solución a los actuales problemas que presenta este proceso en dicha Universidad y por consiguiente la falta de control sobre el mismo. Analizando factores que influyen en el cumplimiento de la guardia como la repetición de turnos, días y postas; a su vez se identifican una serie de reglas a tener en cuenta que pueden dificultar o complejizar esta labor cuando es llevada a cabo por una persona lográndose una planificación donde no se repitan consecutivamente los días, turnos y postas de guardia y se satisfagan las necesidades básicas para el cumplimiento de la misma, tanto para estudiantes y trabajadores involucrados en este proceso. Es independiente del sistema operativo donde se ejecute y presenta un requerimiento de hardware mínimo (14).

Sistema para la Gestión de la Guardia Obrera-Estudiantil en la Facultad 3 (SisGest GOE-Fac3)

Es un sistema de gestión para la planificación ágil y más organizada de la Guardia Obrera-Estudiantil, el mismo permite la gestión de procesos que tributan a un adecuado desempeño, organización y funcionamiento que posibilitó de forma más eficiente el manejo y gestión de la información de dicha actividad con una mayor rapidez en su ejecución. El mismo brinda la posibilidad de gestionar la planificación de la guardia en la facultad 3, donde el sistema es capaz de generar el proceso de planificación de la Guardia Obrera-Estudiantil a través de una serie de requisitos y restricciones.

Realiza cambios o permutas de guardias entre estudiantes, siempre notificando a través del correo
 Por otro lado se registran las incidencias ocurridas en el transcurso de la guardia día por día. (15)

Valoración de los sistemas informáticos analizados

A continuación, se definen una serie de indicadores para la comparación de los sistemas informáticos analizados. Indicadores que se escogen por su repercusión en el desarrollo de la aplicación, debido a que el nivel de reutilización de estos sistemas para llevar a cabo la solución, dependen en gran medida de los mismos:

- Requerimientos de hardware.
- Código abierto.
- Documentación técnica.
- Adaptable al escenario de la Guardia Obrera Estudiantil en la Facultad 3.
- Tecnología libre.

Se propone una ponderación según las necesidades para el desarrollo de la aplicación (entre 1 y 5); lo que significa que para un valor alcanzado de 5, el indicador se encuentra favorecido, en dependencia del indicador en un 100%, para un valor alcanzado de 4, el indicador se encuentra favorecido en un 75% y así sucesivamente. Por ejemplo, si el indicador Requerimiento de Hardware está ponderado con un valor de 5 significa que las prestaciones de la computadora deben estar a un 100%, servicios de internet a toda hora, calidad absoluta del disco duro, RAM (Memoria de Acceso Aleatorio), microprocesador, etc. Si el indicador Código Abierto tiene valor de 1 significa que al no ser un software distribuido y desarrollado libremente no se tiene acceso al código fuente, por lo tanto, no puede reutilizarse el código. Si el indicador Adaptable al escenario de la Guardia Obrera Estudiantil se pondera con un valor de 3 significa que solo el 50% de la aplicación se puede reutilizar o adaptar sus requerimientos y características a las necesidades actuales, es decir, sus tecnologías y herramientas de desarrollo, el marco de trabajo utilizado, los requerimientos funcionales pueden utilizarse como apoyo para lograr el objetivo propuesto. Si el indicador Tecnología Libre se encuentra ponderado con un valor de 4 significa que el 75% de la aplicación se desarrolló con tecnología que tiene licencia libre, que los usuarios de una forma adecuada puedan ejecutar, copiar, distribuir, estudiar el software y que pueda ser distribuido comercialmente a bajo costo.

Los indicadores de mayor nivel de importancia para el desarrollo del nuevo módulo son:

- Código Abierto.
- Adaptable al escenario de la Guardia Obrera Estudiantil en la Facultad 3.

Indicadores	Requerimientos de Hardware	Código Abierto	Documentación Técnica	Adaptable al escenario de la Guardia Obrera Estudiantil en	Tecnología libre
Software				Guardia Obrera Estudiantil en	

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

				la Facultad 3	
AutoMaT P.S	5	1	2	1	1
GCP	5	1	2	1	1
TecnoHospital	5	1	2	1	1
CGO-UCI	3	4	3	2	4
CGEC-Fac7	3	4	3	3	4
SisGes-DM	3	4	4	3	4
PGOE-UCI	3	4	4	4	4
SisGest GOE-Fac3	3	4	4	3	4

Tabla 1: Comparación entre los sistemas informáticos. (Fuente: Elaboración Propia)

Según la tabla ilustrada anteriormente, se llega a la conclusión de que las aplicaciones internacionales estudiadas no cumplen con los indicadores establecidos, sobre todo con los que mayor nivel de importancia tienen para el desarrollo de la aplicación, destacándose el uso de tecnología privativa lo cual dificulta la reutilización tanto del código como de las tecnologías y herramientas de desarrollo, impidiendo su adaptabilidad a las características actuales de la Guardia Obrera Estudiantil, destacando también la escasa documentación técnica para consultar información. A diferencia de estas aplicaciones internacionales, el estudio de los sistemas nacionales según la tabla ilustrada arrojó que la mayoría de los indicadores establecidos son favorables para el desarrollo de la solución. Son aplicaciones multiplataforma, por lo que su despliegue y uso es posible por parte de todos los usuarios que interactúan con la aplicación, en diferentes sistemas operativos. Presentan requerimientos de hardware mínimo, brindan información sobre el control y planificación de la Guardia Obrera Estudiantil y se tiene acceso al código fuente de las mismas.

A pesar de que los sistemas manejan escenarios comunes, gestionan y planifican dependiendo de sus características propias, actualmente ningún sistema está en funcionamiento y no todos cumplen con las características actuales para la planificación, ejecución y control de la Guardia Obrera Estudiantil debido a que estos no gestionan las observaciones y permutas las cuales permiten al personal involucrado realizar cambios en la planificación correspondiente a su guardia, además existen problemas en la realización de los planes de la misma ya que no reflejan concesiones establecidas respecto a: rotaciones por turnos, afinidad de las parejas de guardias, repetición de turnos, días y postas. Por otro lado, está la necesidad por parte del Vicedecanato de Economía y Administración la integración de este módulo con otros módulos y por tanto la utilización de otras herramientas, marcos de trabajo y metodologías que no utilizaban las aplicaciones descritas

anteriormente. Por lo que es más factible la realización de una nueva aplicación web que responda apropiadamente a estas necesidades.

1.4 Metodología de desarrollo de software

De nada sirven buenas notaciones y herramientas si no se proveen directivas para su aplicación. Así, esta década ha comenzado con un creciente interés en metodologías de desarrollo. Dos esquemas se destacan en esta área, el "tradicional" y el "ágil". (16). El primer esquema para abordar el desarrollo de software ha demostrado ser efectivo y necesario en proyectos de gran tamaño (respecto a tiempo y recursos), donde por lo general se exige un alto grado de ceremonia en el proceso. Sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, muchos equipos de desarrollo se resignan a prescindir del "buen hacer" de la ingeniería del software, asumiendo el riesgo que ello conlleva. En este escenario, las metodologías ágiles emergen como una posible respuesta para llenar ese vacío metodológico. Por estar especialmente orientadas para proyectos pequeños 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. (16)

Antes de resumir la metodología escogida para el desarrollo del proyecto, se enumeran las principales diferencias respecto a las metodologías tradicionales ("no ágiles") y las ágiles. (16).

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

Tabla 2: Comparación entre las metodologías. (Fuente:(16))

La metodología escogida para guiar todo el proceso de desarrollo de software fue AUP (Agile Unified Process por su siglas en inglés lo que significa Proceso Unificado Ágil en español) específicamente la variación de AUP para la UCI. La elección se sustenta en los intereses del cliente, dado que estos giran en torno al uso exclusivo de esta metodología. Por otro lado el módulo forma parte de un sistema que podría pertenecer en un futuro a uno de los centros de producción de la UCI, los cuales se encuentran en el programa de mejora en que está inmersa la universidad, lo que trae consigo el

uso de esta metodología. A continuación se enuncian y describen las principales características de las mismas.

AUP (Agil Unified Process)

El Proceso Unificado Ágil de Scott Ambler, es una versión simplificada del Proceso Unificado de Rational (RUP). Describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. Aplica técnicas ágiles incluyendo:

- Desarrollo dirigido por pruebas.
- Modelado ágil.
- Gestión de cambios ágiles.
- Refactorización de base de datos para mejorar la productividad. (17)

Variación de AUP para la UCI.

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.), exigiéndose así que el proceso sea configurable de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Se decide entonces hacer una variación de la metodología AUP permitiendo tanto el desarrollo ágil como el tradicional y en dependencia de las características de cada proyecto genera solo los artefactos principales que se necesiten. (17)

Fases (Variación AUP-UCI)	Objetivos de las fases (Variación AUP-UCI)
Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto.
Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y libera el producto.
Cierre	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Tabla 3: Descripción de las fases AUP variación para la UCI. (Fuente:(17))

A continuación se representan las disciplinas de la metodología Variación AUP para la UCI:

1. Modelado de negocio.
2. Requisitos.
3. Análisis y diseño.
4. Implementación.

5. Pruebas Internas, de Liberación y Aceptación.
6. Gestión de la configuración.
7. Planeación de proyecto.
8. Monitoreo y control de proyecto.
9. Despliegue.

Entre las técnicas ágiles que utiliza la Variación de AUP para la UCI se encuentra el Modelado ágil, se hará uso de esta técnica para los proyectos que necesiten por sus características encapsular sus requisitos funcionales en Historias de usuarios. (17)

La técnica mencionada anteriormente es aplicada por el equipo de desarrollo para la descripción de los requisitos funcionales.

1.5 Tecnologías y herramientas de desarrollo

Para el cumplimiento de los objetivos de la investigación, teniendo en cuenta las características del producto que se quiere lograr, se hizo indispensable hacer una descripción de las herramientas y las tecnologías implicadas en el desarrollo del Sistema de Administración y Economía de la Facultad 3, debido a que se quiere lograr una integración de varios módulos para que formen parte del sistema, trayendo consigo que se utilicen en la solución del trabajo las mismas para todos los módulos. A continuación, se describen las tecnologías y herramientas de desarrollo que se utilizarán para realizar la propuesta de solución:

Lenguaje de modelado: UML 2.0

Se utiliza UML en su versión 2.0 como lenguaje de modelado, ya que el mismo permite al equipo de desarrollo visualizar, especificar, construir y documentar los artefactos del sistema. Esto proporciona una forma estándar de escribir los planos del mismo y cubrir tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables. (18)

Herramienta case: Visual Paradigm 8.0

Es una herramienta que facilita y permite al equipo de desarrollo visualizar, diseñar, integrar y distribuir la aplicación que se necesita. Esto se logra mediante la creación de modelos UML, el modelado de base de datos lo que proporciona una mayor documentación de la misma y diagramas de mapeo de relación de objetos, el modelo de procesos de negocios lo que permite la visualización, improvisación y entendimiento de los procesos. Además ayuda a crear un modelo de excelencia durante la creación y distribución del proceso de desarrollo de aplicaciones. (19)

Lenguaje de programación: PHP 5.4.4

Es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML². Es muy rápido, pues permite al equipo de desarrollo la generación dinámica de páginas, su integración con las bases de datos y el servidor Apache. PHP es multiplataforma y funciona tanto para Unix como para Microsoft Windows, de manera que el código creado para Unix no sufre modificaciones al ser ejecutado en Windows y viceversa. Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. Debido a su amplia distribución y su característica de código abierto, PHP está perfectamente soportado por una gran comunidad de desarrolladores, lo que propicia una rápida reparación de los fallos de funcionamiento que sean detectados. (20)

Marco de trabajo: Symfony2 versión 2.6.3

El marco de trabajo o framework permite simplificar el desarrollo de un sistema mediante la utilización de distintos tipos de lenguajes, el uso de componentes, librerías y estructuras de acuerdo a los diferentes patrones existentes, aporta agilidad y facilidad a los programadores pues encapsula operaciones complejas en instrucciones sencillas y posee una organización bien estructurada del proyecto informático en desarrollo. Symfony2 es un marco estable para desarrollar aplicaciones, está publicado bajo una licencia de software libre y es fácil de instalar y configurar en la mayoría de las plataformas. Es sencillo de usar en la mayoría de los casos y al mismo tiempo es flexible como para adaptarse a los casos más complejos. De acuerdo a las necesidades del sistema y el lenguaje seleccionado, este marco es el ideal para el desarrollo. Fundamentado en que Symfony2 ha sido ideado para exprimir al límite todas las nuevas características de PHP y por eso es uno de los frameworks PHP con mejor rendimiento. Su arquitectura interna está completamente desacoplada, lo que permite adaptarse fácilmente a los diferentes proyectos. Posee la ventaja de ser multiplataforma y compatible con la mayoría de gestores de bases de datos como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Hace uso del MVC (Modelo, Vista, Controlador) y sigue la mayoría de las mejores prácticas y patrones de diseño para la web. (21) (22)

Gestor de base de datos: PostgresSql 9.1.14

² HTML: HyperText Markup Language, por sus siglas en español significa Lenguaje de Marcado de Hipertexto.

Es un servidor de base de datos relacional libre. Permite al equipo de desarrollo la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Cuenta con una gran comunidad de desarrollo en Internet, su código fuente está disponible sin costo alguno y es multiplataforma. La característica fundamental que tiene es:

- Alta concurrencia: mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en español) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases. (23)

Herramienta para administrar el gestor de base de datos: pgAdmin 1.14.2

Es la herramienta de código abierto de administración para la base de datos PostgreSQL. Algunas de las características por las que el equipo de desarrollo la utiliza son: edición rápida de consultas, soporte para todos los tipos de objetos de PostgreSQL. Está empaquetado con el instalador de Windows, y puede ser usado como un cliente para administrar un servidor remoto en otro sistema operativo. Es el centro de administración para las bases de datos PostgreSQL, incluyendo una interfaz gráfica de administración, una herramienta para el trabajo con SQL, un editor de código de procedimientos y funciones, y mucho más. Está diseñado para darle respuesta de las necesidades de la mayoría de los usuarios, desde la escritura de consultas simples en SQL hasta el desarrollo de bases de datos complejas. La interfaz gráfica soporta todas las características presentes de PostgreSQL y se puede hacer la administración fácilmente. Está disponible en más de 30 lenguajes y para varios sistemas operativos. (24)

Entorno de Desarrollo Integrado (IDE): PhpStorm 8.0

Es un programa potente para el desarrollo en PHP que permite la finalización inteligente de código, comprobación de errores al instante. Está disponible para todas las plataformas (Windows, Mac OS X y Linux). Ofrece diferentes niveles de soporte para lenguajes como PHP, JavaScript y HTML con respecto a la edición, la inspección, la refactorización, la extensibilidad a través de plugins. Es adaptable y fácil de usar ya que el programa de edición de PHP entiende perfectamente su estructura, soporta varias versiones y permite abrir ficheros individualmente y editarlo. (25).

Servidor de aplicaciones: Apache 2.2.22

Apache es un servidor web HTTP (Hypertext Transfer Protocol, por sus siglas en español Protocolo de Transferencia de Hipertextos) de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Fundación (Fundación de Software Apache). Presenta entre

otras características altamente configurables, bases de datos de autenticación, negociado de contenido, modular, es de código abierto y multi-plataforma. (26)

Controlador de versiones: Subversion 1.6.17

Para el control de versiones según las necesidades del proyecto se usa Subversion, conocido también como SVN por ser el nombre de la herramienta utilizada en la línea de comando. Las ventajas por las que se selecciona esta herramienta son:

- Utiliza la historia de los archivos y directorios a través de copias y renombrados.
- Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos.
- Puede ser servido mediante Apache.
- Permite selectivamente el bloqueo de archivos, lo que permite que no sean editados por más de una persona a la vez. (27)

Herramienta para el control de versiones: RapidSVN 0.12.0

RapidSVN es un cliente gráfico para Subversion, un programa de control de versiones sustituto de CVS. Es fácil de usar, tanto por quienes ya conocen Subversion como para quien empieza, pudiendo acceder a direcciones SVN, subir y descargar contenido y sincronizarlo con el servidor original, comprobar su estado, crear y fusionar direcciones, etc. Tiene la utilidad de funcionar en varias plataformas y dispone de un completo manual en línea. (28)

Navegador web: Mozilla Firefox 35.0.1

Es un navegador web libre y de código abierto, lo que quiere decir que todo desarrollador puede modificar el código y así poder mejorarlo. Se trata de un navegador que se encuentra disponible en diferentes versiones y es multiplataforma, es decir, es posible utilizarlo con Microsoft Windows, Mac OS X y con GNU/Linux, etc.

Características:

- Navegación por pestañas.
- Corrector ortográfico.
- Búsqueda progresiva.
- Marcadores dinámicos.
- Administrador de descargas.

- Bloqueo de ventanas emergentes no deseadas incorporado buscadores en la barra de herramientas.
- Navegación privada.
- Es un software libre e integración del motor de búsqueda que desee el usuario.

1.6 Conclusiones del capítulo

- Se expusieron los elementos teóricos necesarios para adquirir el conocimiento requerido del tema en cuestión.
- El estudio y comparación de los sistemas de gestión de guardia existentes tanto a nivel nacional como internacional demostró la necesidad de desarrollar una nueva aplicación que satisfaga los requerimientos necesarios para lograr una buena organización, concepción y desarrollo de los procesos de la Guardia Obrera Estudiantil.
- Se realizó la descripción de las metodologías, herramientas y tecnologías de desarrollo lo que permitió concluir que las mismas servirán de sustento al desarrollo de la solución para el cumplimiento de los objetivos propuestos.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.1 Introducción

En este capítulo se describe la propuesta de solución de la aplicación a desarrollar, el modelo conceptual que refleja las entidades que están presentes en el negocio y cómo se relacionan. También se identifican y describen los procesos de negocio, los requisitos funcionales y no funcionales, las técnicas de captura y validación de requisitos. Además se representa el diseño arquitectónico a emplear en el desarrollo de la solución, el diagrama de clases del diseño según el marco de trabajo que se utiliza y patrones de diseño. Se realiza la validación del diseño mediante el uso de diferentes métricas. Por otra parte, se elabora el modelo de implementación el cual está representado por los estándares de codificación, el diagrama de componentes y despliegue para una mayor visualización de la interacción con el sistema y el correcto funcionamiento de sus nodos.

2.2 Propuesta de solución

Debido a las dificultades existentes en la realización de los procesos relacionados con la Guardia Obrera Estudiantil de la Facultad 3, se pretende desarrollar como parte del Sistema de Administración y Economía de dicha facultad un módulo que permita erradicar dichos problemas, para de este modo lograr que se lleven a cabo la gestión de los procesos en un menor tiempo. El módulo brindará la posibilidad de gestionar los procesos de la Guardia Obrera Estudiantil. Para esto el mismo será capaz de generar el plan de la GOE en un determinado rango de fecha teniendo en cuenta una serie de requisitos y restricciones que son de suma importancia para lograr una buena planificación y la satisfacción del personal involucrado en la misma. Además proporcionará la posibilidad de realizar cambios o permutas de guardias entre las personas que posean la misma categoría, lo cual facilitará una mayor flexibilidad y conformidad de acuerdo a las necesidades de las personas que realizan la Guardia Obrera Estudiantil. Por otra parte permitirá el registro de las incidencias ocurridas en un determinado turno en el transcurso de la guardia para un mejor control en la ejecución de la misma.

2.3 Fase de Ejecución de la metodología seleccionada

A continuación se describen las actividades requeridas para desarrollar el software en esta fase. Durante la ejecución se modela el negocio, obtienen los requisitos, se elabora la arquitectura, el diseño, se implementa, valida y libera el producto. La validación y liberación de la solución propuesta se expone en el Capítulo 3.

2.3.1 Modelado del negocio

Durante la fase de modelado del negocio se examina la estructura del proyecto, se observan las entidades del mismo y sus relaciones, se describen los procesos principales y el flujo de trabajo de

cada actividad. Las relaciones y actividades quedan representadas en diagramas simples que indican cuáles son las responsabilidades de cada persona dentro del flujo de trabajo, ayudan a probar que cada uno tenga una visión común de los procesos del negocio y de los roles que existen dentro de ellos. (29). Su objetivo fundamental es comprender el conjunto de actividades que tienen lugar dentro del Vicedecanato de Economía y Administración, fundamentalmente en la gestión de los procesos de la Guardia Obrera Estudiantil, como paso previo a establecer los requisitos del sistema a desarrollar.

Modelo conceptual

El modelo conceptual tiene como objetivo identificar y explicar los conceptos significativos en el problema, identificando los atributos y las asociaciones entre ellos. Es una representación de las entidades, ideas, conceptos u objetos del mundo real. Es un diccionario visual de conceptos e información del dominio del problema. A continuación, se muestra el modelo conceptual del escenario de la Guardia Obrera Estudiantil, donde se evidencian las entidades presentes y las relaciones entre ellas.

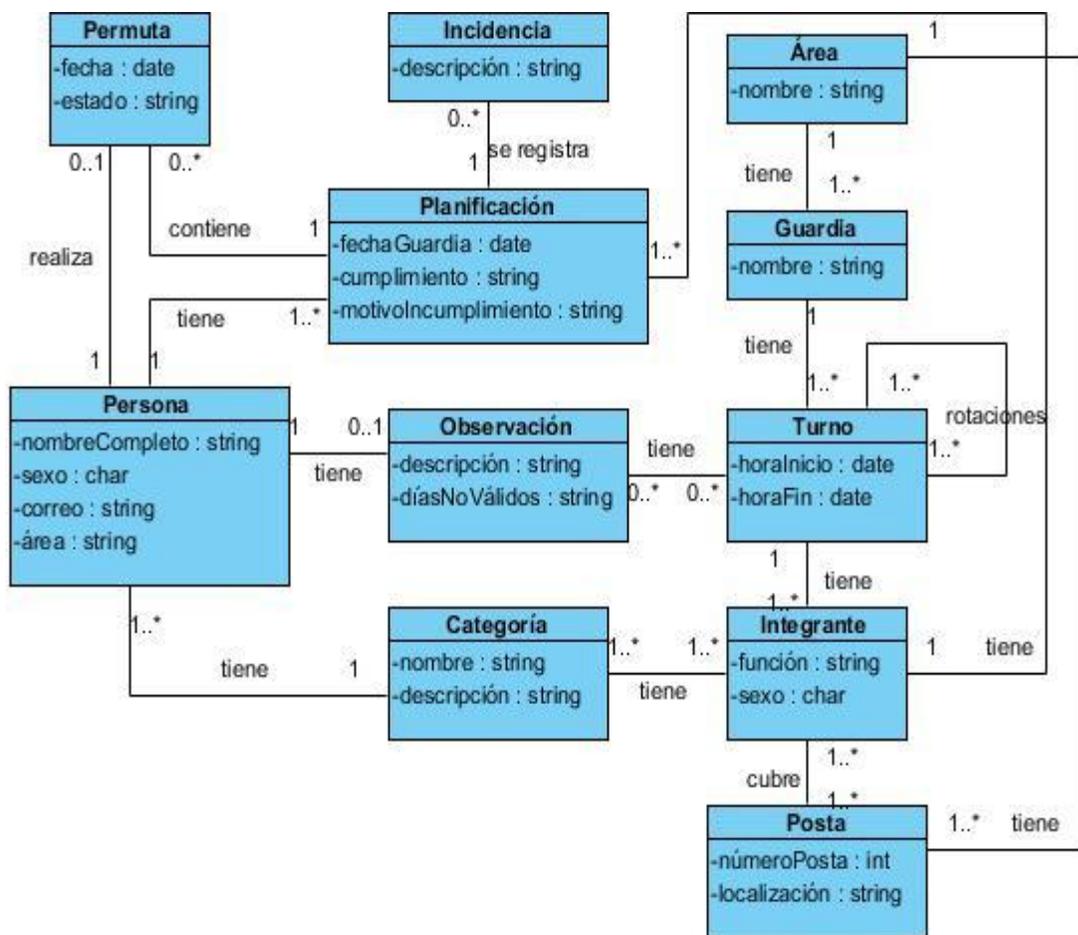


Figura 1: Modelo Conceptual. (Fuente: Elaboración Propia)

Diccionario de datos:

Área: representa un lugar donde se realizará una Guardia Obrera Estudiantil determinada, ya sea docente, residencia, etc.

Planificación: representa la planificación y el cumplimiento de la Guardia Obrera Estudiantil, especificando los motivos por los cuales la persona no cumplió con la planificación de la misma.

Guardia: representa el nombre del tipo de guardia que se va a realizar.

Permuta: representa los cambios que puede realizar una persona en la planificación correspondiente a su Guardia Obrera Estudiantil.

Categoría: describe la categoría que puede tener una persona, por ejemplo si es estudiante, trabajador, etc.

Incidencia: describe los problemas ocurridos en la ejecución de la Guardia Obrera Estudiantil.

Persona: refleja toda la información necesaria respecto a determinada persona para la planificación la Guardia Obrera Estudiantil, puede ser estudiante o trabajador.

Observación: representa una breve descripción de la observación realizada a las personas que participan en la Guardia Obrera Estudiantil y por tanto los días y turnos que no puedan realizar.

Turno: refleja los turnos que se van a realizar en el desarrollo de la Guardia Obrera Estudiantil, su comienzo y fin y las rotaciones por turno que se tendrán en cuenta a la hora de planificarle a las personas.

Integrante: representa a cada integrante de la Guardia Obrera Estudiantil, la categoría y función que desempeñan en el desarrollo de la misma.

Posta: representa el lugar exacto, es decir, una localización más específica de donde se realizará la Guardia Obrera Estudiantil.

Para más información sobre las entidades del negocio (Ver anexos del 1 al 11).

Procesos de negocio

Un proceso de negocio, permite visualizar el objetivo del proceso, las entradas, salidas y el flujo de eventos del mismo, ya sea el flujo básico que es cuando todo ocurre de manera satisfactoria hasta que se culmina el proceso. También permite identificar los flujos alternos que son las actividades que se realizan cuando el proceso no se desarrolla satisfactoriamente y los flujos paralelos que son cuando se realizan dos actividades conjuntas en el proceso.

El equipo de desarrollo en conjunto con la clienta, dependiendo de las características y necesidades actuales existentes en el negocio identificó los siguientes procesos: planificar Guardia Obrera

Estudiantil, ejecutar Guardia Obrera Estudiantil y controlar Guardia Obrera Estudiantil. En los diagramas de procesos de negocio se hará uso de las siglas GOE para referirse a la Guardia Obrera Estudiantil debido a lo extenso del nombre. A continuación se representa el diagrama de proceso de negocio planificar Guardia Obrera Estudiantil, donde las actividades que se representan en un color rosado son las actividades a informatizar en el módulo GOE.

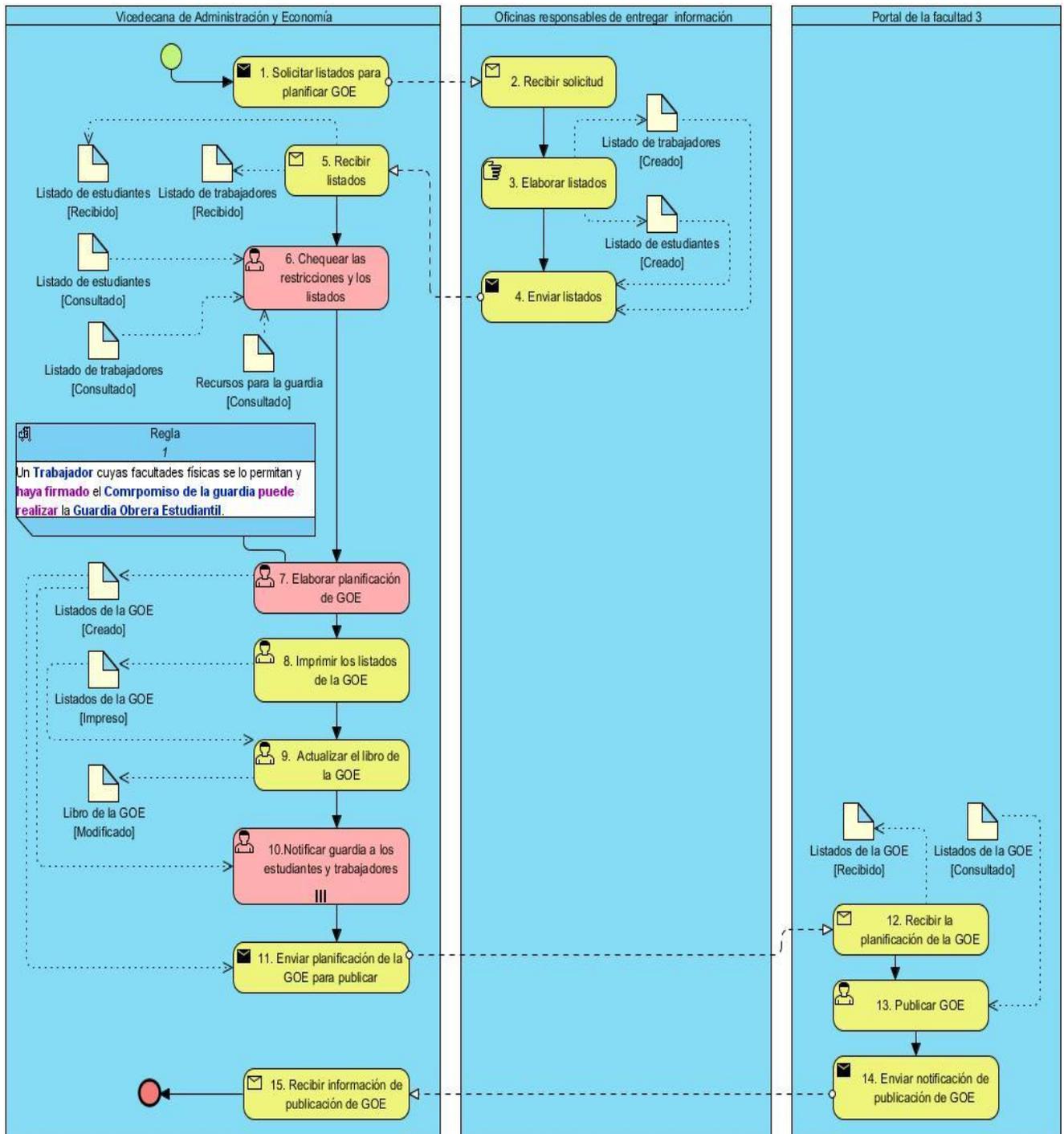


Figura 2: Diagrama de proceso: Planificar GOE. (Fuente: Elaboración propia)

CAPÍTULO II: PROPUESTA DE SOLUCIÓN

Objetivo	Realizar la planificación de la Guardia Obrera Estudiantil de la Facultad 3.
Evento(s) que lo genera(n)	Solicitar la información para planificar la Guardia Obrera Estudiantil.
Pre condiciones	Debe existir el documento con la firma de los trabajadores comprometidos para realizar la Guardia Obrera Estudiantil, es decir el excel Recursos para la guardia.
Marco legal	N/A.
Reglas de negocio	<ol style="list-style-type: none"> 1. Realizar la Guardia Obrera Estudiantil todos los trabajadores cuyas facultades físicas se lo permitan y hayan firmado el compromiso de la Guardia Obrera Estudiantil. 2. Realizar la Guardia Obrera Estudiantil todos los estudiantes cuyas facultades físicas se lo permitan. 3. Tener en cuenta las rotaciones que debe haber entre los turnos. 4. Planificar a las mujeres en el primer turno o segundo del docente y tener al menos un varón en cada turno. 5. Tener en cuenta la afinidad entre las personas y las observaciones que se le realicen. 6. Planificar a los trabajadores la guardia solo dos fines de semana en siete meses.
Responsable	Vicedecana de Economía y Administración.
Clientes internos	N/A.
Clientes externos	N/A.
Entradas	Recursos para la guardia
Flujo de eventos	
Flujo básico Planificar GOE	
1.	Solicitar listados para planificar GOE: La Vicedecana de Economía y Administración solicita a Secretaría y lo Jefes de Área de los trabajadores la información necesaria para poder planificar GOE, ya sea listado de trabajadores y de estudiantes, además cuenta con el Excel de Recursos para la guardia.
2.	Recibir solicitud: Los Jefes de Área responsables de los trabajadores y el personal de Secretaría reciben la solicitud para buscar la información para la Vicedecana de Economía y Administración.
3.	Elaborar listados: Los Jefes de Área responsables de los trabajadores y el personal de Secretaría reciben la solicitud para buscar la información para la Vicedecana de Economía y Administración y elaborar los listados.
4.	Enviar listados: Las trabajadoras de Secretaría y los Jefes de Área responsables de los trabajadores buscan la información deseada y crean los listados de estudiantes, y trabajadores para enviárselo a la Vicedecana de Economía y Administración.
5.	Recibir listados: La Vicedecana de Economía y Administración recibe la información necesaria para poder planificar la GOE.
6.	Chequear restricciones y los listados: La Vicedecana de Economía y Administración chequea algunas restricciones como por ejemplo, que el trabajador haya firmado el compromiso de la guardia, si hay trabajadoras embarazadas, algún estudiante tiene un problema físico y no puede realizar la guardia, en fin comprueba todas las reglas del negocio. Luego actualiza los listados.
7.	Elaborar planificación de GOE: La Vicedecana de Economía y Administración planifica la GOE teniendo en cuenta que no se repitan los turnos, las postas y los horarios, el personal disponible, las restricciones antes mencionadas.

CAPÍTULO II: PROPUESTA DE SOLUCIÓN

8. Imprimir los listados de la GOE: La Vicedecana de Economía y Administración imprime los listados con los estudiantes, trabajadores, responsables de cada turno, posta, día en que realizan la guardia, etc.

9. Actualizar el libro de la GOE: La Vicedecana de Economía y Administración adjunta la planificación de la GOE de ese mes al libro de guardia que contiene la GOE de todos los meses.

10. Notificar guardia a los estudiantes y trabajadores: Se les envía a los estudiantes, profesores y trabajadores que sus condiciones físicas le permitan hacer la guardia mediante el correo la planificación de la GOE.

11. Enviar planificación con la GOE para publicar: La Vicedecana de Economía y Administración envía la planificación de la GOE a un encargado del sitio dragones.uci.cu para que publiquen la guardia y cualquier inconveniente los implicados puedan ver esto desde el sitio.

12. Recibir planificación con GOE: El encargado de publicar la guardia en el sitio dragones.uci.cu recibe el Excel con la planificación.

13. Publicar planificación de la GOE: El encargado en el sitio publica la planificación de la GOE.

14. Enviar notificación de publicación de GOE: El encargado en el sitio le dice a la Vicedecana de Economía y Administración que ya se publicó la GOE.

15. Recibir información de publicación de GOE: La Vicedecana de Economía y Administración se informa que la planificación de la GOE ya se publicó.

16. Concluye el proceso.

Pos-condiciones

1. Se obtiene un Excel con la planificación de la GOE de la facultad 3.

Salidas

1. Listado impreso con la planificación de la GOE de la facultad 3.

Flujos paralelos

N/A

Pos-condiciones

1. N/A

Salidas

1. N/A

Flujos alternos

1. N/A

Pos-condiciones

1. N/A

Salidas

1. N/A

Asuntos pendientes

N/A

Tabla 4: Descripción del proceso de negocio: Planificar GOE.

(Fuente: Elaboración Propia)

Para más información de los restantes procesos de negocio y sus descripciones (Ver anexos 12 al 14)

2.3.2 Ingeniería de Requisitos

Los requerimientos o requisitos como bien se conoce para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Los requerimientos reflejan las necesidades de los clientes que ayude a resolver algún problema. El proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requerimientos. Dentro de la ingeniería de requerimientos se encuentran los requerimientos de usuario que no son más que declaraciones y diagramas de los servicios que se espera el sistema proporcione y de las restricciones bajo las cuales deba funcionar y los requerimientos del sistema que establecen con detalle las funciones, servicios y restricciones optativas. Los mismos se clasifican en funcionales y no funcionales. (30)

Existen diferentes técnicas para la captura de requisitos. Las utilizadas por el equipo de desarrollo se especifican a continuación.

Técnicas de captura de requisitos:

- Entrevistas: se le aplicó a la Vicedecana de Economía y Administración permitiendo al equipo de desarrollo obtener información con la mayor calidad posible e interpretar las necesidades del cliente, identificando así los requisitos funcionales a los que debe responder el módulo GOE. (30)
- Tormenta de ideas: se realizó entre el equipo de desarrollo, algunos analistas y desarrolladores del proyecto CEDRUX y el cliente, donde se identificaron los requisitos funcionales a los que debe responder el módulo GOE mediante la intervención participativa. (30)

Requisitos Funcionales (RF)

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar la aplicación, de la manera en que debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. Describen lo que el sistema debe hacer. (30). Mediante las técnicas de captura de requisitos se identificaron un total de 47 RF. A continuación, se muestra un listado de los mismos, agrupados en dependencia del objetivo o servicio que ofrecerán dentro del módulo.

RF_1 Gestionar planificación de la GOE

RF_1.1 Planificar GOE.

RF_1.2 Modificar planificación de GOE.

RF_1.3 Eliminar planificación de GOE.

RF_1.4 Buscar planificación de GOE.

RF_1.5 Eliminar por fecha.

RF_2 Exportar planificación de la GOE.

RF_3 Notificar planificación de GOE.

RF_4 Borrar planificación de la persona.

RF_5 Anular áreas por fecha.

RF_6 Gestionar turnos de la GOE

RF_6.1 Adicionar turno.

RF_6.2 Modificar turno.

RF_6.3 Eliminar turno.

CAPÍTULO II: PROPUESTA DE SOLUCIÓN

RF_7 Gestionar postas de la GOE

RF_7.1 Adicionar posta.

RF_7.2 Modificar posta.

RF_7.3 Eliminar posta.

RF_8 Gestionar control de la GOE

RF_8.1 Generar reportes.

RF_8.2 Buscar.

RF_8.3 Insertar observación de ausencia.

RF_8.4 Modificar observación de ausencia.

RF_9 Controlar ausencias.

RF_10 Gestionar GOE

RF_10.1 Adicionar GOE.

RF_10.2 Modificar GOE.

RF_10.3 Eliminar GOE.

RF_11 Realizar permuta de GOE.

RF_12 Confirmar permuta de GOE.

RF_13 Cancelar permuta de GOE.

RF_14 Gestionar equipo por turno

RF_14.1 Adicionar integrante.

RF_14.2 Modificar integrante.

RF_14.3 Eliminar integrante.

RF_15 Gestionar persona

RF_15.1 Adicionar persona.

RF_15.2 Modificar persona.

RF_15.3 Eliminar persona.

RF_15.4 Buscar persona.

RF_15.5 Adicionar observación.

RF_15.6 Modificar observación.

RF_15.7 Insertar afinidad.

RF_15.8 Modificar afinidad.

RF_15.9 Eliminar por área.

RF_16 Importar listado de personas.

RF_17 Gestionar categoría

RF_17.1 Adicionar categoría.

RF_17.2 Modificar categoría.

RF_17.3 Eliminar categoría.

RF_18 Gestionar incidencia

RF_18.1 Adicionar incidencia.

RF_18.2 Modificar incidencia.

RF_18.3 Eliminar incidencia.

RF_19 Gestionar grupos de rotación

RF_19.1 Buscar rotación por turno.

RF_19.2 Adicionar rotación por turno.

RF_19.3 Modificar rotación por turno.

Requisitos No Funcionales

Los RNF son restricciones que no se refieren directamente a las funciones ofrecidas por el sistema, sino a las propiedades emergentes de este como restricciones de tiempo de respuesta, estándares, fiabilidad, capacidad de almacenamiento. (30). A continuación, se muestra un listado de los requisitos no funcionales definidos por el equipo de desarrollo en conjunto con el cliente y agrupados según los tipos de requisitos no funcionales existentes:

RNF	Nombre	Número	Descripción
Requerimientos del Producto	Usabilidad	1	El sistema debe representar un diseño sencillo, lo que permite que no sea necesario mucho entrenamiento para que los usuarios puedan utilizar el sistema en un período de

			dos días.
		2	El sistema debe poseer una interfaz agradable para el cliente y fácil de operar.
	Fiabilidad	3	El sistema debe estar disponible 24 horas del día, los siete días de la semana.
		4	El sistema debe tener soporte para recuperación ante fallos y errores. Esto se logrará mediante la herramienta de implementación a utilizar.
	Soporte	5	El sistema debe tener soporte para grandes volúmenes de datos y velocidad de procesamiento.
		6	El sistema debe tener un tiempo de respuesta rápido en accesos concurrentes.
Requerimientos externos	Seguridad	7	El sistema debe brindar protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
		8	El sistema debe garantizar la confidencialidad, integridad y disponibilidad de la información que se procese en el sistema.
		9	El control de acceso se establecerá por roles que se le asignarán a los usuarios que interactúen con el sistema.
Requerimientos organizacionales	Restricciones de diseño	10	El sistema utiliza el lenguaje de programación es php.
		11	El sistema utiliza los marcos de trabajo symfony2, bootstrap, etc.
		12	El sistema utiliza la plataforma de desarrollo phpStorm.
		13	El sistema utiliza la herramienta case para modelado visual paradigm.
		14	El sistema utiliza la herramienta gestora de base de datos postgresSQL.

Tabla 5: Requisitos No Funcionales. (Fuente: Elaboración Propia)

Descripción de las historias de usuario

Siguiendo la metodología seleccionada se realizará la descripción de los RF mediante las historias de usuario. En la tabla 6, se especifica el requisito funcional Planificar GOE. Para más información sobre las restantes historias de usuario de los demás RF (Ver anexos del 15 al 21).

Número: RF_1.1	Nombre del requisito: Planificar GOE
Programador: Luis Angel Castillo Rodriguez	Iteración Asignada: 2

Prioridad: Alta	Tiempo Estimado: 21
Riesgo en Desarrollo: Alto	Tiempo Real: Tres semanas
Descripción: Inicia cuando la Vicedecana de Economía y Administración decide planificar la guardia a los estudiantes y trabajadores de la Facultad 3. Para esto necesita el listado de los estudiantes y trabajadores, de postas y turnos con los que dispone y definir un rango de tiempo a planificar.	
Observaciones: N/A	
Prototipo de interfaz:	
	

Tabla 6: Historia de Usuario: Planificar GOE.

(Fuente: Elaboración Propia)

Existen diferentes técnicas para la validación de requisitos. Las utilizadas por el equipo de desarrollo se especifican a continuación.

Técnicas de validación de requisitos

- Revisiones técnicas formales: los requisitos fueron analizados sistemáticamente por el equipo de desarrollo y el cliente en busca de anomalías y/u omisiones.
- Prototipos: se utilizó para mostrar un modelo ejecutable del sistema a los usuarios finales y los clientes, para analizar si dicho modelo cumple con las necesidades reales del escenario de la Guardia Obrera Estudiantil.
- Generación de casos de prueba: se realizó un caso de prueba para cada requisito, lo que permitió detectar errores o defectos relacionados con las funcionalidades del módulo y para comprobar la veracidad del código.

2.3.3 Análisis y diseño

Una de las disciplinas de la fase de Ejecución que propone la metodología Variación AUP para la UCI es el análisis y el diseño. El mismo es un enfoque de la ingeniería de software que modela un sistema como un grupo de objetos que interactúan entre sí. Este enfoque representa un dominio absoluto en

términos de conceptos compuestos por verbos y sustantivos, clasificados de acuerdo a su dependencia funcional. En este método de análisis y diseño se crea un conjunto de modelos utilizando una notación acordada como, por ejemplo, el Lenguaje Unificado de Modelado (UML). Aplica técnicas de modelado de objetos para diseñar soluciones que mejoren los procesos involucrados y para analizar los requerimientos en diferentes contextos, por ejemplo, un sistema de negocio, un conjunto de módulos de software, etc. (29)

Diseño Arquitectónico

El diseño arquitectural permite organizar diferentes puntos de vista y controlar el desarrollo de un sistema mediante la selección y evaluación de los aspectos estructurales para cumplir con los requisitos funcionales, como de los aspectos no estructurales para cumplir con los requisitos no funcionales como por ejemplo: el rendimiento, la capacidad de adaptación y la reutilización. (31) El diseño de la aplicación está basado en el patrón MVC (Modelo–Vista–Controlador).

Patrón arquitectónico MVC

El patrón MVC, es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. (32)

Este patrón tiene un modelo, vistas y en ocasiones varios controladores. Las vistas y los controladores suelen estar muy relacionados ya que tratan los eventos que se producen en la interfaz gráfica (vista). Esta separación de aspectos de una aplicación da mucha flexibilidad al desarrollador. (33)

Vista: Presenta el modelo (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho modelo la información que debe representar como salida. Es la página HTML. (33)

Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). Son código que obtienen datos dinámicamente y genera el contenido HTML. (33)

Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también

los privilegios de acceso, dicha información es almacenada en una base de datos o en XML junto con las reglas de negocio que transforman esa información (teniendo en cuenta las acciones de los usuarios). (33).

Symfony2 basa su funcionamiento interno en el patrón arquitectónico Modelo-Vista-Controlador (MVC), utilizado por la mayoría de frameworks web. (34). En la figura 4 se describe el funcionamiento de este patrón arquitectónico en el marco de trabajo utilizado.

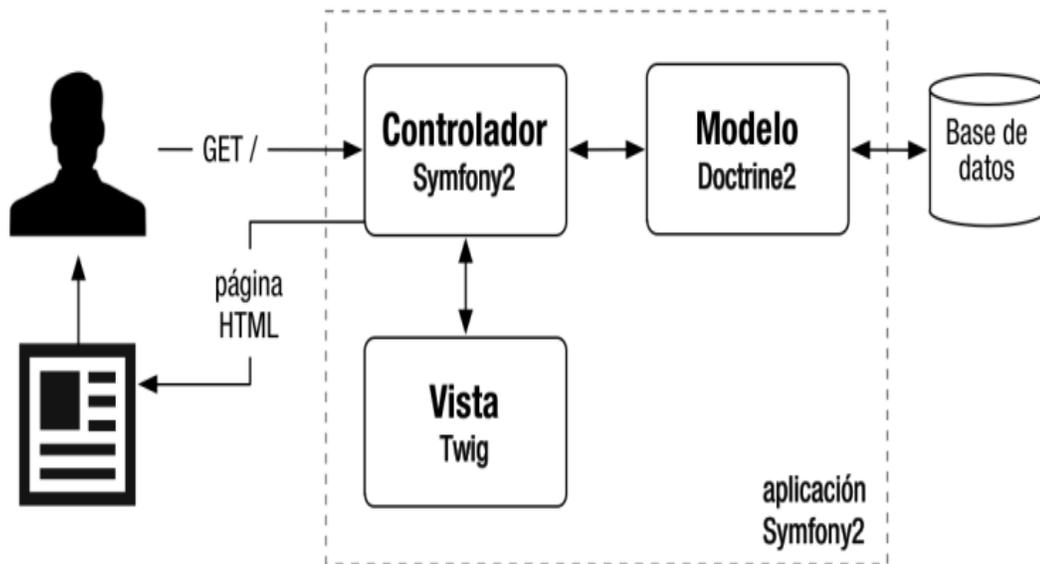


Figura 3: Funcionamiento general del patrón MVC.

(Fuente:(34))

Cuando el usuario solicita ver la portada del sitio, internamente sucede lo siguiente:

- El sistema de enrutamiento determina qué Controlador está asociado con la página de la portada, luego Symfony2 ejecuta el Controlador asociado a esa portada y a su vez el Controlador solicita al Modelo los datos necesarios, este se los devuelve, el Controlador solicita la Vista que cree una página mediante una plantilla y que inserte los datos del Modelo y por último el Controlador entrega al servidor la página creada por la Vista.(34)

Diagrama de clases del diseño

El diagrama de clases con estereotipos web o diagrama de clases del diseño muestran la especificación para las clases software de la aplicación. Incluye información como clases,

CAPÍTULO II: PROPUESTA DE SOLUCIÓN

asociaciones, atributos, interfaces con sus operaciones y constantes, métodos, dependencias, muestra definiciones de entidades de software más que conceptos del mundo real, se añaden los detalles referentes al lenguaje de programación que se vaya a usar. Representado mediante tres capas: la capa de presentación, la capa del negocio, la capa de acceso a datos con una relación entre ellas. La capa de presentación tiene como estereotipos principales la Server Page las cuales permiten construir y controlar la información de las páginas clientes que serían las Client Page y estas muestran información en el cliente manejando los datos del formulario que contiene toda esa información. En la capa del negocio se muestran las Clases Controladoras, las cuales controlan todo el funcionamiento de la aplicación y las Clases Entidades que contienen todos los datos necesarios de las entidades existentes. Por último la capa de acceso a datos que permite el acceso a esos datos mediante una clase Repositorio. A continuación, se representa uno de los diagramas de clases de diseño realizados "Gestionar planificación de la GOE". El mismo fue escogido por la importancia del escenario de la planificación en la Guardia Obrera Estudiantil. (Ver anexos del 22 al 27).

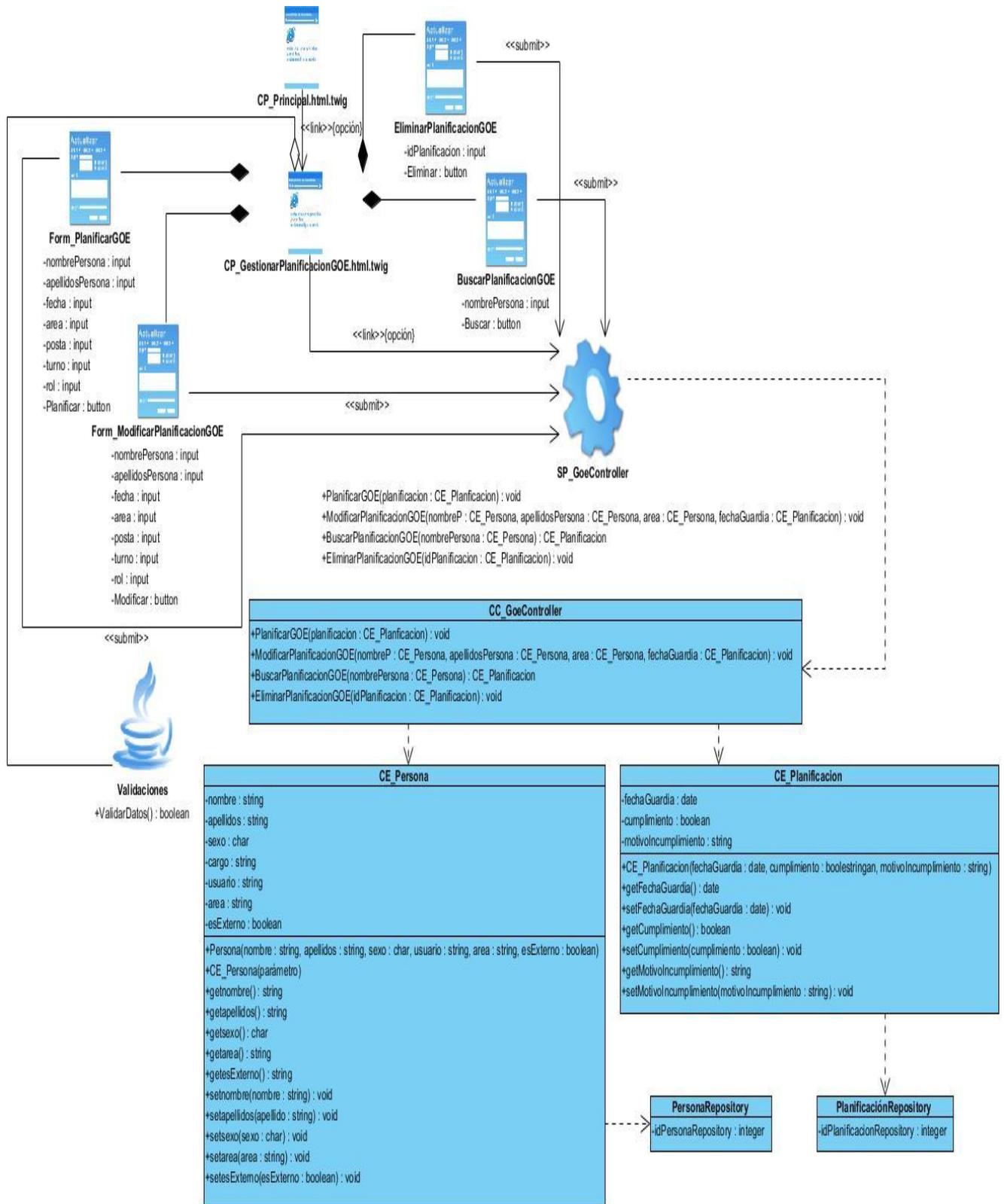


Figura 4: Diagrama de clases del Diseño: Gestionar planificación de la GOE.

(Fuente: Elaboración Propia)

Patrones de diseño

Un patrón de diseño es una solución a un problema de diseño que se presenta en el desarrollo de un software. Entre sus características se debe mencionar que es reutilizable y aplicable a diferentes problemas de diseño en distintas circunstancias. Es una descripción de un problema y la solución que brinda y que se puede aplicar a nuevos contextos. (35). Para resolver la asignación de responsabilidad en las clases del diseño el equipo de desarrollo hizo uso de los Patrones Generales de Software para asignar Responsabilidades.

➤ Patrones GRASP

General Responsibility Assignment Software Patterns (GRASP), Patrones Generales de Software para asignar Responsabilidades; describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. (35). En el trabajo de diploma se utilizaron los siguientes patrones de diseño:

Experto: este patrón es aplicado en todas las clases debido a que cada una de ellas es experta pues contienen la información necesaria para cumplir con las responsabilidades que le fueron asignadas; facilitando así el entendimiento y contribuir al bajo acoplamiento. (29). En la clase entidad Planificación se evidencia el uso de este patrón. (Ver figura 6)

Creador: el patrón se evidencia en las clases controladoras que, para cada una de las funcionalidades de la aplicación, son las encargadas de crear las instancias de los objetos que manejan, favoreciendo así la reutilización y el bajo acoplamiento. (29). La clase GOE Controller es la encargada de crear las instancias de los objetos que maneja. (Ver figura 8)

Bajo Acoplamiento: se proporciona un bajo acoplamiento en el diseño debido a que las clases existentes tienen asignadas responsabilidades de tal forma que estas no dependen en gran medida de otras clases. (29). En la figura 7 se evidencia el uso de este patrón.

Alta Cohesión: el patrón se evidencia debido a que a cada una de las clases se le asignaron responsabilidades de tal forma, que están estrechamente relacionadas entre sí y no realizan un trabajo excesivo. (29). En la figura 7 se evidencia el uso de este patrón.

Una de las características principales de Symfony es la organización del trabajo en cuanto a la estructura del proyecto, lo cual permite crear y trabajar con clases con una alta cohesión. Por ejemplo, las clases con sufijo Controller en su nombre contienen varias funcionalidades estrechamente relacionadas entre ellas, teniendo un sentido común y un propósito único, siendo las encargadas de controlar las acciones de las plantillas y por lo tanto pertenecen a la capa del Controlador dentro de la arquitectura Modelo-Vista-Controlador. Esto hace posible que el software sea flexible a cambios sustanciales con efecto mínimo.



Figura 5: Patrón Experto. (Fuente: Elaboración propia)

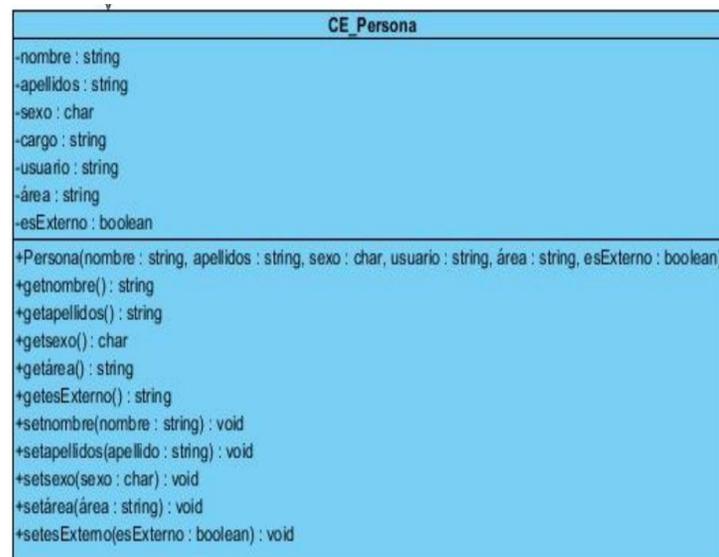


Figura 6: Patrón Bajo Acoplamiento y Alta Cohesión. (Fuente: Elaboración propia)

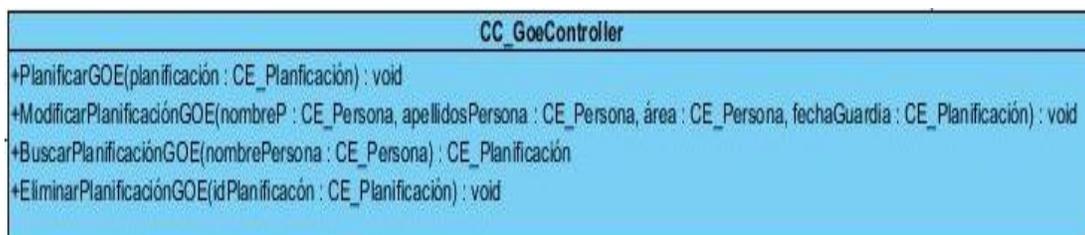


Figura 7: Patrón Creador. (Fuente: Elaboración propia)

Validación del diseño propuesto

Las métricas permiten medir de forma cuantitativa la calidad de los atributos internos del software. (36). Esto permite al equipo evaluar la calidad durante el desarrollo del sistema.

Para la evaluación de la calidad del diseño propuesto se utilizaron las métricas básicas inspiradas en la calidad del diseño orientado a objeto, en la cual se abarcan atributos de calidad que permiten medir la calidad del diseño propuesto. A continuación, se describen los mismos:

- **Responsabilidad:** consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- **Cantidad de pruebas:** consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad del producto diseñado.
- **Complejidad de implementación:** consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
- **Complejidad de mantenimiento:** consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costos y la planificación del proyecto.
- **Reutilización:** consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.
- **Acoplamiento:** consiste en el grado de dependencia o interconexión de una clase o estructura de clase con otras, está muy ligada a la característica de reutilización. (36)

Las métricas descritas para evaluar la calidad del diseño y su relación con los atributos de calidad definidos son las siguientes y se encuentran explicadas y representadas en el expediente de proyecto mediante el artefacto **Métricas de Diseño GOE**.

➤ **Métrica Tamaño Operacional de Clase (TOC)**

Se refiere al número de métodos pertenecientes a una clase. Está determinada por los atributos: responsabilidad, complejidad de implementación y la reutilización, existiendo una relación directa con los dos primeros e inversa con la última. A continuación se describen los atributos que la integran y la forma de calcular la dimensión en que estos atributos afectan el diseño propuesto. (36)

Atributo que afecta	Modo en que lo afecta
reutilización	Un aumento del TOC implica una disminución en el grado de reutilización de la clase.
complejidad de implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.

Tabla 7: Tamaño Operacional de Clase (TOC). (Fuente: Elaboración Propia)

En la tabla 8 se encuentran representados los umbrales que se utilizan en la métrica TOC para definir la categoría de los atributos utilizados.

Atributos	Categoría	Criterio
reutilización	Baja	$> 2 * Prom.$
	Media	Entre Prom. y $2 * Prom.$
	Alta	$\leq Prom.$
complejidad de implementación	Baja	$\leq Prom.$
	Media	Entre Prom. y $2 * Prom.$
	Alta	$> 2 * Prom.$
responsabilidad	Baja	$\leq Prom.$
	Media	Entre Prom. y $2 * Prom.$
	Alta	$> 2 * Prom.$

Tabla 8: Rango de valores para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de implementación y Reutilización) relacionados con la métrica TOC.

Resultados del instrumento de evaluación de la métrica TOC.

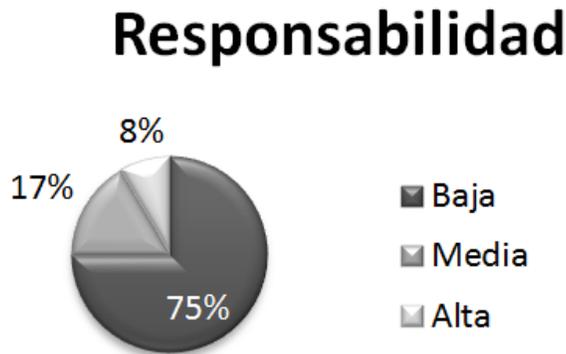


Figura 8: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.

Complejidad

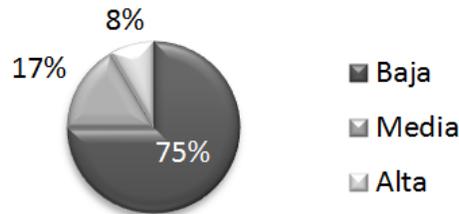


Figura 9: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación.

Reutilización

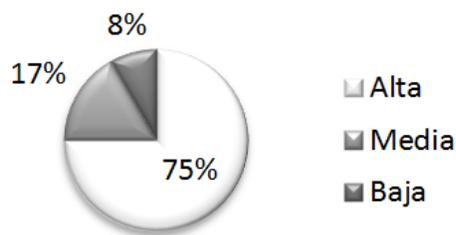


Figura 10: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.

Al analizar los resultados obtenidos luego de aplicar el instrumento de medición de la métrica TOC, se puede concluir que los tres atributos de calidad se encuentran en un nivel satisfactorio en el 75% de las clases, de manera que se puede observar cómo se fomenta la reutilización (elemento clave en el proceso de desarrollo de software) y cómo se reducen la responsabilidad y la complejidad de implementación, por lo que el diseño propuesto para el sistema es simple y tiene calidad.

➤ Métrica Relaciones entre Clases (RC)

Está dada por el número de relaciones de uso de cada clase. Se determina por los atributos: acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización, existiendo una relación directa con los tres primeros e inversa con el último antes mencionado. A continuación se describen los atributos que la integran y la forma de calcular la dimensión en que estos atributos afectan el diseño propuesto.(36)

Atributo que afecta	Modo en que lo afecta
reutilización	Un aumento del RC implica una disminución en el grado de reutilización de la

CAPÍTULO II: PROPUESTA DE SOLUCIÓN

	clase.
complejidad de mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
acoplamiento	Un aumento del RC implica un aumento del Acoplamiento de la clase.
cantidad de pruebas	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

Tabla 9: Relaciones entre clases. (Fuente: Elaboración propia)

En la tabla 11 se encuentran representados los umbrales que se utilizan en la métrica RC para definir la categoría de los atributos utilizados.

Atributo	Categoría	Criterio
acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
complejidad de mantenimiento	Baja	< =Prom
	Media	Entre Prom. y 2* Prom
	Alta	> 2* Prom
reutilización	Baja	> 2*Prom
	Media	Entre Prom. y 2* Prom
	Alta	<= Prom
cantidad de pruebas	Baja	< =Prom
	Media	Entre Prom. y 2* Prom
	Alta	> 2* Prom

Tabla 10: Rango de valores para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas) relacionados con la métrica RC.

Resultados del instrumento de evaluación de la métrica RC

Reutilización

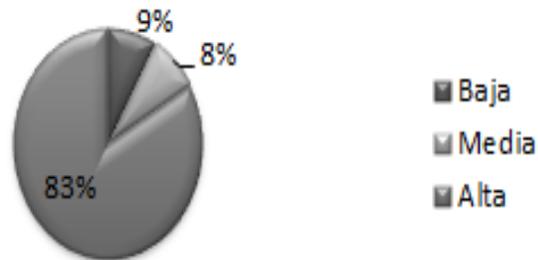


Figura 11: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización.

Complejidad de Mantenimiento

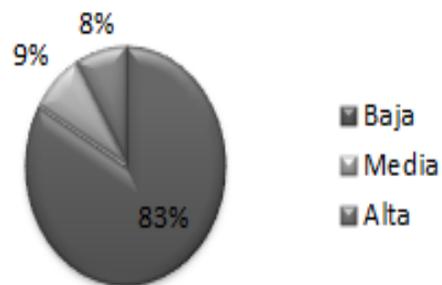


Figura 12: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de mantenimiento.

Acoplamiento

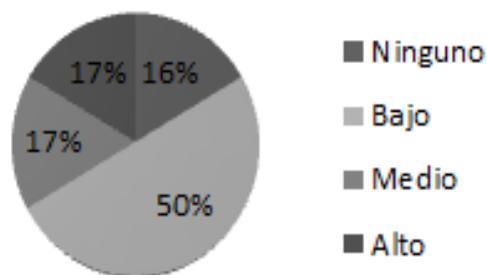


Figura 13: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.

Cantidad de Pruebas

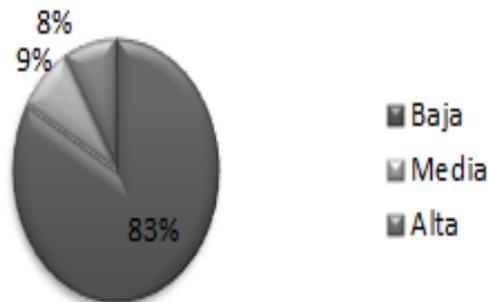


Figura 14: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas.

Al analizar los resultados obtenidos luego de aplicar el instrumento de medición de la métrica RC, se puede concluir que el diseño propuesto para el sistema es simple y tiene calidad ya que los atributos de medición se encuentran en un nivel satisfactorio. En el 58% de las clases el acoplamiento es bajo, lo que implica que el grado de dependencia entre las clases es mínimo. La complejidad de mantenimiento se comporta favorablemente para un 83% de las clases donde el grado de esfuerzo necesario para desarrollar una mejora o una rectificación de algún error en el diseño de software es mínimo. Para un 83% de las clases el atributo cantidad de pruebas es bajo por lo que grado de esfuerzo para realizar las pruebas de calidad del producto diseñado es mínimo. El grado de reutilización presente en el diseño es alto debido a que el atributo reutilización se encuentra favorecido para un 83% de las clases.

Modelo de datos

Uno de los artefactos generados en la disciplina de Análisis y diseño es el modelo de datos. En la figura 16 se representa el modelo de datos de la propuesta de solución. El mismo está compuesto por un total de 15 tablas donde se describen las estructuras de datos, su tipo, la forma en que se relacionan y las restricciones de integridad entre estas tablas. (37)

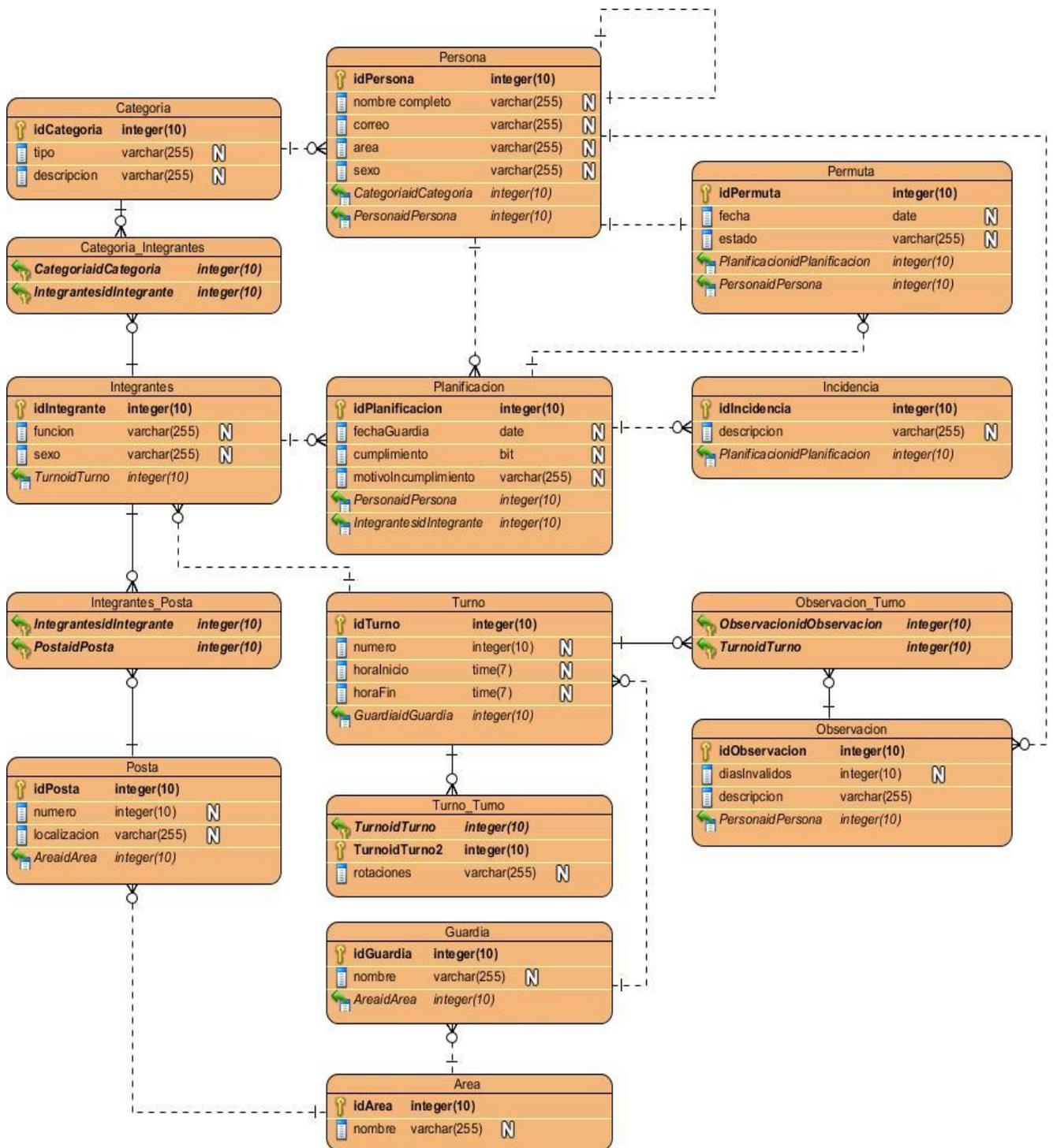


Figura 15: Modelo de datos. (Fuente: Elaboración Propia)

2.3.4 Modelo de implementación

Como resultado del tránsito por la disciplina de implementación se genera el modelo de implementación, el mismo se realiza con el objetivo de transformar la estructura de la disciplina

Análisis y diseño en un programa ejecutable. (38). Para que una implementación sea legible y fácil de mantener es necesario definir estándares a seguir por el equipo de desarrollo.

Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código el cual refleja un estilo armonioso, como si un único programador hubiese escrito todo el código de una sola vez. Cuando el proyecto de software incorpora código fuente previo, o cuando realiza el mantenimiento de un sistema de software el estándar de codificación debería establecer cómo operar con la base de código existente. La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. La mantenibilidad del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento. (38)

Pautas a seguir para el trabajo con las clases del módulo GOE

- Las clases controladoras comienzan con el nombre seguido de la palabra Controller (GoeController.php).
- Las clases entidades relacionadas en el modelo con las clases Repository de Symfony2 comienzan con el nombre de la entidad seguido de la palabra Repository (PlanificacionRepository).
- Las clases entidades para diferenciarlas de otros módulos en el sistema se escribirán con la palabra goe delante.
- El código de las clases .php será colocado dentro de la carpeta src\AppBundle\Controller\goe. El nombre de estas clases comenzará con mayúscula.
- Se utilizan los comentarios en caso de ser necesario para explicar la utilidad de la clase.
- Los nombres de las clases se escriben con la primera letra de cada palabra que lo compone en minúscula y las palabras que le siguen con la primera letra en mayúscula.
- Los nombres de clases deben ser sustantivos y deben tener la primera letra en mayúsculas. Si el nombre es compuesto, cada palabra componente deberá comenzar con mayúsculas. Los nombres serán simples y descriptivos. Debe evitarse el uso de acrónimos o abreviaturas, salvo en aquellos casos en los que dicha abreviatura sea más utilizada que la palabra que representa (URL, HTTP, etc.).

```
class PlanificacionGoe {
```

```
...
```

}

Pautas a seguir para nombrar las variables y métodos

- Los nombres deben ser descriptivos y concisos. No usar grandes frases y solo abreviaturas pequeñas para el caso de GOE.
- Las variables se escribirán siempre en minúsculas. Las variables compuestas tendrán la primera letra de cada palabra componente en mayúsculas.

Ejemplo: `private $adicionarGuardia`

- El identificador para las variables y los parámetros serán con letras en minúsculas y en caso de ser un nombre compuesto se divide cada palabra con mayúscula inicial.
- Los métodos deben ser verbos escritos en minúsculas. Cuando el método esté compuesto por varias palabras cada una de ellas tendrá la primera letra en mayúsculas.

Ejemplo: `public function planificarGuardia()`**Pautas a seguir para la división de líneas**

Cuando una expresión ocupe más de una línea, esta se podrá romper o dividir en función de los siguientes criterios:

- Tras una coma.
- Antes de un operador.
- Se recomienda las rupturas de nivel superior a las de nivel inferior.
- Alinear la nueva línea con el inicio de la expresión al mismo nivel que la línea anterior.

Ejemplos:

```
unMetodo(expresionLarga1, expresionLarga 2, expresionLarga 3,
expresionLarga 4, expresionLarga 5);
```

```
if ((condicion1 && condicion2)
```

```
|| (condicion3 && condicion4)
```

```
||!(condicion5 && condicion6)) {unMetodo(); }
```

Pautas a seguir para la declaración de clases / interfaces

- Durante el desarrollo de clases / interfaces se deben seguir las siguientes reglas de formateo:

- No incluir ningún espacio entre el nombre del método y el paréntesis inicial del listado de parámetros.
- El carácter inicio de bloque ("{") debe aparecer al final de la línea que contiene la sentencia de declaración.
- El carácter fin de bloque ("}") se sitúa en una nueva línea tabulada al mismo nivel que su correspondiente sentencia de inicio de bloque, excepto cuando la sentencia sea nula, en tal caso se situará detrás de "{".
- Los métodos se separarán entre sí mediante una línea en blanco.

```
class ClaseEjemplo extends OtraClase {  
  
    $variable1;  
  
    $variable2;  
  
    public function __construct() {  
  
        $variable1 = 0;  
  
        $variable2 = 1;  
  
    }  
  
    ...  
  
}
```

Diagrama de componentes

Los diagramas de componentes representan cómo un sistema de software es dividido en componentes y las dependencias entre ellos, los cuales son utilizados para modelar la vista estática y dinámica de un sistema lo que permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que los componentes proporcionan y utilizan a través de las interfaces. (38).

A continuación, se representa el diagrama de componentes correspondiente al módulo GOE, el cual está dividido en tres paquetes, el Modelo, la Vista y el Controlador y por otros componentes que utiliza internamente Symfony2, ejemplo Configuraciones, Controlador Frontal, Seguridad y Rutas, en donde el Controlador Frontal hace uso de los restantes componentes mencionados. Dentro del Modelo se encuentra el componente Entidades del que hace uso el componente Repositorio para el acceso a la base de datos. En la Vista se destaca el componente Twig que es un sistema de plantillas que utiliza el marco de trabajo, el mismo hace uso de los componentes Formularios y Extensiones. En

el Controlador el componente Controladoras hace uso de los componentes Entidades del Modelo y de los Formularios y las plantillas Twig en la Vista.

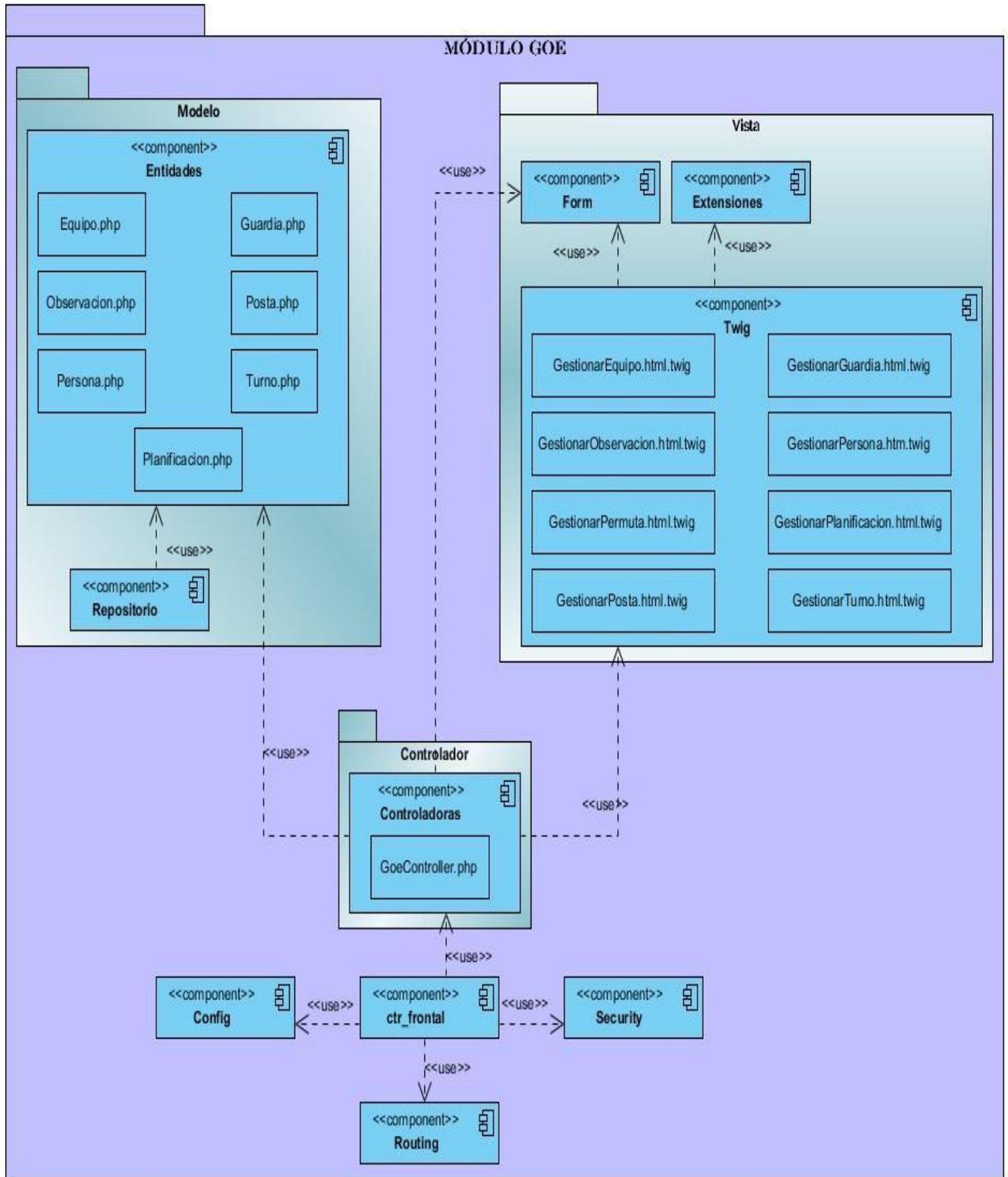


Figura 16: Diagrama de componentes. (Fuente: Elaboración Propia)

Diagrama de despliegue

El diagrama de despliegue UML se muestra la arquitectura física de un sistema informático, se representan los equipos, dispositivos, sus interconexiones y el software que se encontrará en cada máquina. (38).

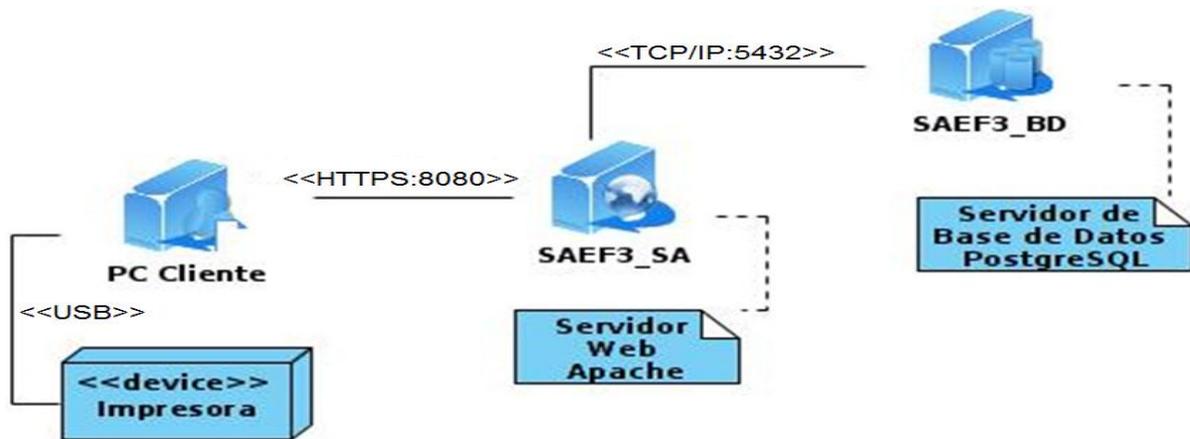


Figura 17: Diagrama de despliegue. (Fuente: Elaboración Propia)

En la figura 18 se muestra cómo será desplegado el SAEF3 al cual pertenece el módulo GOE. El diagrama de despliegue está compuesto por la conexión de la PC_Cliente y un dispositivo Impresora mediante el bus de serie universal (USB) o haciendo uso del protocolo de control de transmisión y el protocolo de internet (TCP/IP), mientras que la misma se conecta al servidor de aplicaciones web Apache a través del protocolo de transferencia de hipertexto (HTTPS) por el puerto 8080 y con el servidor de Base de Datos PostgreSQL mediante el puerto 5432.

2.4 Conclusiones del capítulo

- Se creó el modelo conceptual, así como los diagramas de los procesos de negocio y sus descripciones lo cual permitió una mejor comprensión de todas las especificidades y el funcionamiento de cada proceso a informatizar.
- Se diseñó el modelo de datos, permitiendo establecer una estructura de almacenamiento para toda la información relacionada con los procesos de gestión de la Guardia Obrera Estudiantil, mostrándose además las relaciones existentes entre las diferentes tablas de la base de datos.
- Se crearon los diagramas de clases del diseño permitiendo conocer la estructura y las relaciones de las clases que se manejan en dicho sistema para alcanzar un entendimiento más profundo de cómo se realizará la implementación del sistema, realizando también la propuesta del diseño fue validada lo cual arrojó como resultado que la aplicación puede ser reutilizada, de fácil mantenimiento y baja complejidad algorítmica.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

3.1 Introducción

En el presente capítulo se muestran los resultados obtenidos de las pruebas realizadas a la aplicación web mediante la estrategia de pruebas. La misma se divide en niveles de prueba y cada nivel tiene métodos de prueba. (29). Se explica su funcionamiento y resultados a través de las pruebas realizadas al software para detectar errores o defectos relacionados con sus funcionalidades y para comprobar la veracidad del código.

3.2 Estrategia de pruebas

Como parte de la fase de ejecución de la metodología seleccionada se valida la propuesta de solución mediante la estrategia de prueba, la misma integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que llevan a la evaluación correcta del software. La estrategia de pruebas de software proporciona un mapa que describe los pasos que se darán para realizarla, indica cuando se planea y se darán dichos pasos, además cuanto tiempo, esfuerzo y recursos consumirán. Un software se prueba para descubrir los errores cometidos, si se realiza sin ningún plan seguramente se desperdicia tiempo, un esfuerzo innecesario y lo que es peor puede que no se detecten los errores. (29)

Las pruebas integran un elemento más amplio el cual se llama verificación y validación. (29). Las estrategias de prueba siguen varios objetivos entre los que se encuentran:

- Planificar las pruebas necesarias en cada iteración, incluyendo las pruebas de unidad, integración, de alto nivel o pruebas de validación y las pruebas de sistema. (29)
- Diseñar e implementar las pruebas creando los casos de prueba que especifican qué probar, cómo realizar las pruebas y creando, si es posible, componentes de prueba ejecutables para automatizar las pruebas. (29)
- Realizar diferentes pruebas y manejar los resultados de cada prueba sistemáticamente. (29)

Los objetivos identificados anteriormente serán evidenciados en el transcurso del capítulo.

3.3 Niveles de prueba

A la hora de evaluar un sistema generalmente se comienza probando las partes más pequeñas y se continúa con las más grandes. El módulo (componente) se prueba primero y luego se continúa con la integración de módulos. (39)(41). Las pruebas se aplican en distintos niveles de trabajo, entre las que se encuentran:



Figura 18: Niveles de Prueba de Software. (Fuente:(41))

- **Pruebas de unidad:** se concentran en probar cada componente individualmente para asegurar que funcione de manera apropiada como unidad. Principalmente se utiliza el método de caja blanca o estructural que suelen llevarse a cabo con el acceso al código fuente, el cual emplea técnicas de prueba que recorren caminos específicos en la estructura de control de los componentes. (39) (40) (41)
- **Pruebas de integración:** Las pruebas de integración tienen dos objetivos principales: descubrir errores asociados con las interfaces de los módulos y ensamblar sistemáticamente los módulos individuales para formar subsistemas y al final un sistema completo. Principalmente se utiliza el método de caja negra o funcional con la técnica de diseño de casos de prueba que verifican el correcto manejo de las entradas y salidas del software. (39) (40) (41)
- **Pruebas de validación:** se enfocan en los requerimientos, se validan los requisitos establecidos como parte del análisis de requisitos del software, comparándolos con el sistema que ha sido construido. (39) (40) (41)
- **Pruebas del sistema:** este nivel es más adecuado para comprobar requisitos no funcionales como seguridad, velocidad, exactitud, fiabilidad etc. Se enfoca en la integración del sistema (Hw, información, personas). (39) (40) (41)

El equipo de desarrollo realizará las pruebas de unidad para probar que cada componente individualmente funciona de manera apropiada, las pruebas de integración para descubrir errores asociados con las interfaces del módulo y las pruebas de aceptación como parte de las pruebas de validación de los requisitos establecidos comparándolos con el sistema que ha sido construido.

3.4 Métodos de prueba

Para realizar las pruebas seleccionadas, el equipo de desarrollo hará uso de los diferentes métodos existentes en cada una de ellas. Los métodos de prueba tienen un enfoque sistemático, independiente del nivel en que se enmarque la prueba, que ayuda a encontrar buenos conjuntos de casos de prueba para detectar diferentes tipos de errores. Existen dos métodos básicos para diseñar casos de prueba: de caja blanca (o estructural) y de caja negra (o funcional). (41)

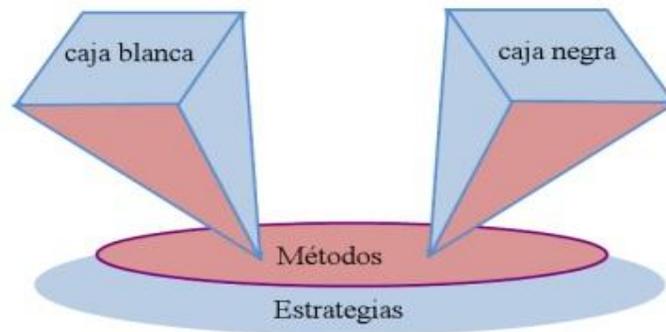


Figura 19: Métodos de prueba caja blanca y negra. (Fuente:(39))

3.4.1 Pruebas de caja blanca o estructural

Las pruebas de caja blanca verifican la correcta implementación de las unidades internas, las estructuras, sus relaciones, y hacen énfasis en la reducción de errores internos. Permite derivar casos de prueba que garanticen que todas las rutas independientes dentro del módulo se ejecuten al menos una vez, que ejecuten los lados verdadero y falso de todas las decisiones lógicas, todos los ciclos dentro y en sus límites operacionales y por último que ejerciten las estructuras de datos internas para asegurar su validez. (39) (40) (41)

Para lograr esto la técnica utilizada por el equipo de desarrollo es la ruta del camino básico la cual analiza un segmento de código y lo transforma en un grafo que contiene nodos, que no es más que la secuencia de procedimientos; las aristas que son el flujo de control y las regiones que son el espacio cerrado comprendido entre el nodo y las aristas, además siempre existe una región externa. Existen tres formas de calcular la complejidad, estas son:

$$V(G) = \text{Aristas} - \text{Nodos} + 2$$

$$V(G) = P + 1$$

$$V(G) = \#R$$

A: aristas.

N: nodos.

P: nodo predicado (cantidad de nodos de donde sale más de una arista).

R: número de regiones.

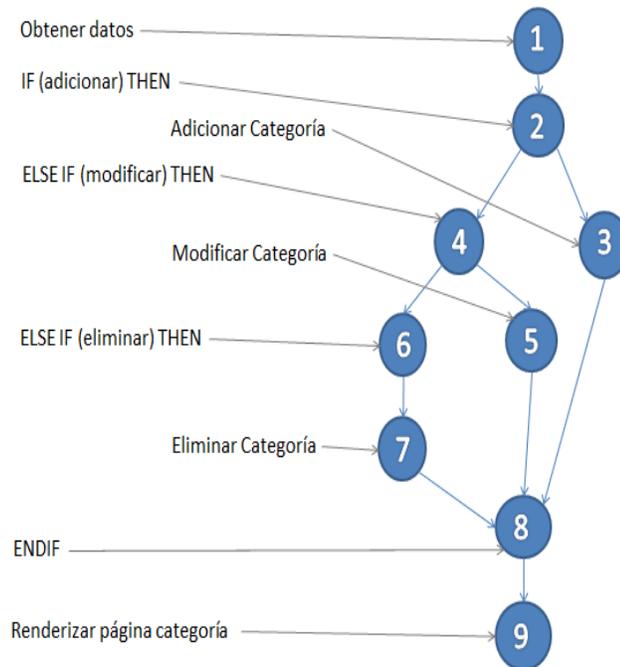


Figura 20: Grafo formado mediante la técnica del camino básico.

(Fuente: Elaboración propia)

3.4.2 Pruebas de caja negra o funcional

Las pruebas de caja negra verifican el correcto manejo de funciones externas provistas o soportadas por el software y que el comportamiento observado se apegue a las especificaciones del producto y a las expectativas del usuario. (39) (40) (41)

La técnica utilizada por el equipo de desarrollo para encontrar errores de funciones incorrectas o faltantes, errores de interfaz, errores en estructuras de datos o en acceso a BD externas, errores de comportamiento o desempeño y errores de inicialización o término es el diseño de casos de prueba. Su objetivo es probar todos los requisitos funcionales del sistema y analizar la respuesta que dan ante una ejecución de los mismos y recopilar las no conformidades obtenidas para su posterior corrección.

3.4.3 Pruebas de aceptación

El equipo de desarrollo en conjunto con la cliente y los tutores realizó las pruebas de aceptación al módulo GOE, haciendo uso de la técnica ladov. La técnica ladov se compone de cinco preguntas claves: tres cerradas y dos abiertas, las cuales se reformulan en la investigación para valorar el grado de satisfacción de los clientes sobre el tema en específico. Una vez establecidas las preguntas se conforma el “cuadro lógico de ladov” y el número resultante de la interrelación de las tres preguntas,

CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN

indica la posición de los sujetos en la escala de satisfacción. (42) La escala de satisfacción está dada por los criterios:

1. Máxima satisfacción.
2. Más satisfecho que insatisfecho.
3. No definida.
4. Más insatisfecho que satisfecho.
5. Máxima insatisfacción.
6. Contradictoria. (42)

Pregunta cerrada 3	Pregunta cerrada 1								
	No			No sé			Sí		
	Pregunta cerrada 2								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho									
No me gusta mucho									
Me da lo mismo									
Me disgusta más de lo que me gusta									
No me gusta nada									
No sé qué decir									

Tabla 11: Cuadro lógico de Iadov. (Fuente: Elaboración propia)

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma:

Índice de satisfacción	Escala
Máxima satisfacción	+1
Más satisfecho que insatisfecho	0,5
No definido y contradictorio	0
Más insatisfecho que satisfecho	-0,5
Máxima insatisfacción	-1

Tabla 12: Índice de satisfacción grupal. (Fuente: Elaboración propia)

La satisfacción grupal (ISG) se calcula por la siguiente fórmula:

$$ISG = \frac{A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)}{N}$$

Dónde:

- A: representa el número de sujetos con índice individual 1.
- B: representa el número de sujetos con índice individual 2.
- C: representa el número de sujetos con índice individual 3 ó 6.
- D: representa el número de sujetos con índice individual 4.
- E: representa el número de sujetos con índice individual 5.
- N: representa el número total de sujetos del grupo.

3.5 Pruebas realizadas al sistema

Para la validación de la implementación de las funcionalidades de la aplicación, el equipo de desarrollo seleccionó la técnica de la ruta del camino básico del método de pruebas de caja blanca. A continuación se presenta el código tomado como base para representar el procedimiento anteriormente explicado. Corresponde al método `gestionarCategoriaAction()`. Este método fue seleccionado porque en él se evidencia el uso del CRUD, una de las razones por la que los requisitos fueron agrupados, reflejando las características de gran parte de los RF. Además se escoge porque los datos de las condiciones de los nodos predicados están adecuadamente establecidos, con el fin de comprobar cada camino.

```

public function gestionarCategoriaAction() {
    extract($_REQUEST);
    $em = $this->getDoctrine()->getManager();
    $categorias = $em->getRepository('AppBundle:Categoria')->findAll();
    if (isset($badicionar)) {
        $categoria = new Categoria();
        $categoria->setNombre($nombre);
        $categoria->setDescripcion($descripcion);
        $em->persist($categoria);
        $em->flush();
    } elseif (isset($bmodificar)) {
        $categoria = $em->getRepository('AppBundle:Categoria')->find($idCategoria);
        $categoria->setNombre($nombre);
        $categoria->setDescripcion($descripcion);
        $em->persist($categoria);
    } elseif (isset($beliminar)) {
        $categoria = $em->getRepository('AppBundle:Categoria')->find($idCategoria);
        $em->remove($categoria);
        $em->flush();
    }
    return $this->render('goe/gestionar_categoria.html.twig',
        array('categorias' => $categorias));
}

```

Figura 21: Segmento de código seleccionado. (Fuente: Elaboración propia)

Grafo de flujo a partir del código mostrado:

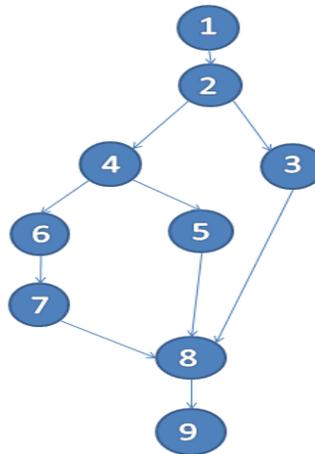


Figura 22: Grafo formado luego de aplicar la técnica del camino básico al código seleccionado.

(Fuente: Elaboración propia)

El grafo fue formado a partir de la aplicación de la técnica del camino básico, lo que permitió la selección de los diferentes caminos que se puede tomar para llegar a una solución. Estos caminos se reflejan en el código mostrado en las condicionales if. A continuación, se procede a calcular la complejidad de código seleccionado para definir el número de caminos linealmente independientes que se necesitan probar en la aplicación.

Complejidad ciclomática:

1. $V(G) = 3$
2. $V(G) = 10 - 9 + 2 = 3$
3. $V(G) = 3$

El valor calculado definió el número de caminos linealmente independientes que fueron probados en el código de la aplicación, donde se obtuvo un resultado satisfactorio de cero errores al comprobar cada uno de ellos. Estos caminos son:

Camino 1: 1-2-3-8-9

Camino 2: 1-2-4-5-8-9

Camino 3: 1-2-4-6-7-8-9

Para verificar el correcto funcionamiento del sistema se realizaron pruebas funcionales aplicando el método de caja negra, y específicamente la técnica de partición de equivalencia. Esta última divide el dominio de entrada de un programa en un número finito de variables de equivalencia. Las variables de equivalencia representan un conjunto de estados válidos y no válidos para las condiciones de entrada de un programa. Se definen dos tipos de variables de equivalencia, las válidas, que representan entradas válidas al programa o las llamadas en los casos de prueba V, y las no válidas,

que representan valores de entrada erróneos o las llamadas en los casos de prueba F. (39) (40). Para cada uno de los requisitos funcionales fueron diseñados los casos de prueba, de los cuales se muestra uno a continuación:

Caso de prueba para el requisito Planificar GOE

Condiciones de ejecución
1- Se debe seleccionar la opción Planificación
2- Debe existir en el sistema al menos una persona, un área, una guardia, un turno, un categoría creada y los integrantes que conformarán cada equipo de guardia.

Figura 23: Condiciones para que se pueda ejecutar la guardia. (Fuente: Elaboración propia)

SC Planificar GOE						
Escenario	Descripción	Guardia	A partir de	Hasta	Respuesta del sistema	Flujo central
EC 1.1: Planificar la GOE introduciendo los datos válidos	Se planifica la GOE introduciendo los datos válidos.	V Guardia Mayo	V 06/30/2015	V 06/30/2015	El sistema planifica la GOE y luego se cierra la interfaz.	<input type="checkbox"/> Se selecciona la opción Planificación. <input type="checkbox"/> Se introducen los datos correctamente. <input type="checkbox"/> Se presiona el botón Planificar.
EC 1.2: Planificar la GOE introduciendo los datos inválidos.	Se planifica la GOE introduciendo los datos inválidos	/ vacío	/ 06/30/2015	/ 06/30/2015	El sistema borra los datos de los campos introducidos incorrectamente y muestra un mensaje de error: "Existen datos incorrectos".	<input type="checkbox"/> Se selecciona la opción Planificación. <input type="checkbox"/> Se introducen los datos para realizar la planificación incorrectamente. <input type="checkbox"/> Se presiona el botón Planificar. <input type="checkbox"/> Se muestra un mensaje de error.

Figura 24: Escenarios del caso de prueba Planificar GOE. (Fuente: Elaboración propia)

Para más información sobre el resto de los casos de pruebas asociados a las restantes funcionalidades (Ver anexos del 28 al 33).

Para lograr que la salida final del sistema informático coincidiera con el resultado esperado por el cliente y para probar cada una de las interfaces de las funcionalidades establecidas en la aplicación se realizaron un total de 64 pruebas funcionales, divididas en tres iteraciones. En la primera iteración se realizaron 20 pruebas funcionales, de las cuales 10 salidas coincidieron con los resultados esperados representando un 50% y 10 pruebas resultaron fallidas. En una segunda iteración se realizaron 37 pruebas funcionales, para los restantes requisitos que se implementaron y para las funcionalidades que resultaron fallidas, de las cuales 30 resultaron satisfactorias representando el 65% y 7 pruebas fallidas. En una tercera iteración se realizaron pruebas a las funcionalidades que resultaron fallidas de la segunda iteración lo que arrojó resultados favorables para un 100% de

CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN

pruebas satisfactorias, lo que significa que el sistema funciona correctamente cumpliendo con las expectativas del cliente.

A continuación, se muestra como quedan reflejados los resultados por cada una de las iteraciones de pruebas funcionales de caja negra realizadas al sistema:

Iteración	Satisfactoria	Insatisfactoria		Total
		Significativas	No significativas	
1	10	10		20
		3	7	
2	30	7		37
		3	4	
3	7	0		7
		0	0	

Tabla 13: Iteraciones de pruebas funcionales.

(Fuente: Elaboración propia)

Para valorar el grado de satisfacción del cliente con la solución desarrollada respecto al control y disponibilidad de la información, se aplicó la técnica ladov que permite el estudio del grado de satisfacción del personal involucrado en un proceso objeto de análisis.

Durante la valoración fungieron como clientes, el consejo de dirección de la facultad 3. A continuación se evidencian los resultados alcanzados luego de aplicar la técnica.

¿Le gustaría hacer uso del módulo GOE del SAEF3 propuesto, para desarrollar los procesos de planificación, ejecución y control asociados a la Guardia Obrera Estudiantil del VDEA?	¿Considera usted oportuno continuar ejecutando los procesos de planificación, ejecución y control asociados a la Guardia Obrera Estudiantil del VDEA manualmente, a pesar del gran volumen de información que se genera?								
	No			No sé			Sí		
	¿El módulo GOE del SAEF3 contribuye a mejorar el control y disponibilidad de la información del VDEA acorde a sus necesidades?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No

CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN

Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta mucho	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	6
No sé qué decir	2	3	6	3	3	3	6	3	4

Tabla 14: Cuadro lógico de Iadov completado. (Fuente: Elaboración propia)

Como resultado se obtuvo:

$$ISG = \frac{5 * 1 + 2 * 0,5}{7}$$

De manera que el ISG = 0,86

Los resultados de la satisfacción individual según las categorías empleadas fueron los siguientes:

Nivel de satisfacción	Cantidad	%
Máxima satisfacción	6	85,71
Más satisfecho que insatisfecho	1	7,14
No definida	0	0

Tabla 15: Resultado de aplicación de la técnica Iadov. (Fuente: Elaboración propia)

Al procesar las respuestas a las encuestas en el cuadro lógico de Iadov, se obtiene un grado de satisfacción grupal de 0,86, lo cual se traduce en una clara satisfacción con el uso del módulo GOE del SAEF3.

En el criterio respecto al control y disponibilidad de la información en el VDEA a través del uso de la solución propuesta, a pesar de que un 7,14% no sabe si sería oportuno continuar ejecutando los procesos asociados al área económica del VDEA manualmente hubo una concordancia de un 100% en que contribuye a su mejora y hacer uso del módulo GOE del SAEF3 para desarrollar los procesos de planificación, ejecución y control asociados a la Guardia Obrera Estudiantil del VDEA .

Las preguntas abiertas que se formularon fueron:

¿Qué valoraciones le sugiere el módulo del SAEF3 respecto al control y disponibilidad de la información asociada al VDEA?

¿Qué elemento(s) usted añadiría a la solución que se propone?

Entre las valoraciones positivas obtenidas como respuestas a las preguntas abiertas, se recopilaron criterios como los siguientes:

- Se señala positivamente las facilidades que brinda el sistema para la visualización de la planificación de la Guardia Obrera Estudiantil.

- Se señala positivamente la posibilidad de realizar permutas entre los usuarios de una misma categoría.
- Se señala positivamente la posibilidad de notificación personal mediante el uso de correo de la planificación de la Guardia Obrera Estudiantil.
- Se señala positivamente la posibilidad de notificación personal de incumplimiento de la planificación de la Guardia Obrera Estudiantil mediante el uso de correo.
- Se subraya como elemento positivo la flexibilidad del módulo GOE para incorporar nuevas funcionalidades.
- Se subraya como elemento positivo la posibilidad de generar reportes de control de la Guardia Obrera Estudiantil.

También se consideró como mejora del módulo, incluir algoritmos basados en la inteligencia artificial para optimizar la planificación de la Guardia Obrera Estudiantil en la facultad.

La aplicación de la técnica de ladov aportó información significativa respecto al grado de satisfacción del cliente. Los resultados obtenidos y los criterios emitidos validan la fortaleza de la propuesta, reflejándose una valoración muy positiva del cliente con la propuesta de solución.

Para más información sobre la validación del módulo GOE por parte del cliente y el consejo de dirección de la facultad (Ver anexo 34).

3.6 Conclusiones del capítulo

- La aplicación de las pruebas de caja negra y pruebas de caja blanca a las funcionalidades y al código permitieron detectar un número significativo de no conformidades, a las cuales se les dio solución obteniendo finalmente una aplicación que satisface los requisitos definidos.
- La aplicación de la prueba de aceptación con el cliente, tutores, el equipo de desarrollo y los usuarios finales que participaron permitió validar que las funcionalidades definidas para resolver los problemas que existían referentes a la Guardia Obrera Estudiantil cumplen las expectativas de los involucrados en la misma.

CONCLUSIONES

Luego de finalizado el presente trabajo de diploma se concluye que se fomentaron todas las tareas de la investigación propuestas con el fin de cumplir los objetivos específicos planteados, resaltando que:

- El estudio y análisis de los principales referentes teóricos y sistemas informáticos tanto nacionales como internacionales asociados a la gestión de la Guardia Obrera Estudiantil, permitió demostrar que no existía una aplicación informática capaz de cubrir todas las funcionalidades requeridas para realizar una adecuada gestión de la misma, por lo que se elaboró y desarrolló un sistema informático que permite realizar una correcta gestión de la Guardia Obrera Estudiantil en la Facultad 3.
- El análisis y el diseño del módulo GOE para la Facultad 3, posibilitó agrupar todas las necesidades del cliente y los usuarios finales que mediarán con la aplicación desarrollada, además de conocer la estructura y las relaciones de las clases que se manejan en dicho módulo para alcanzar un entendimiento más profundo de cómo se realizó la implementación del mismo, de modo que la aplicación puede ser reutilizada, de fácil mantenimiento y baja complejidad algorítmica.
- Con la implementación se obtuvo el módulo GOE de la Facultad 3 perteneciente al Sistema de Administración y Economía de la Facultad 3 (SAEF3) que responde a todas las necesidades, especificidades, flexibilidad que debe brindar la gestión de la Guardia Obrera Estudiantil hoy día y una mejora considerable en la disponibilidad y control de la información.
- La validación de la solución propuesta mediante las pruebas de caja blanca y las pruebas de caja negra permitieron encontrar y corregir los errores no detectados durante la implementación posibilitando cumplir con las especificaciones requeridas y la validación del diseño propuesto, arrojando resultados satisfactorios demostrando la autenticidad de la aplicación desarrollada.

RECOMENDACIONES

Teniendo como base los resultados de este trabajo y la experiencia adquirida durante el desarrollo del mismo, se recomienda:

- Realizar un estudio a nivel de Universidad para incorporar nuevas funcionalidades al sistema de forma tal que pueda ser usado en todas las facultades.
- Realizar un estudio para incorporar nuevas funcionalidades al sistema que permitan generar diferentes reportes.

REFERENCIAS BIBLIOGRÁFICAS

1. Informe de la Administración Guardia Obrera Estudiantil. Facultad 3, Universidad de las Ciencias Informáticas, La Habana, Cuba : s.n., 2013-2014.
2. **Bravo Carrasco, Juan.** Gestión de procesos. Santiago de Chile : Evolución S.A, 2011. ISBN 978-956-7604-20-3.
3. **Horacio Saroka, Raúl.** Sistemas de información en la era digital. Argentina : Fundación OSDE, 2002. ISBN: 987-9358-08-2.
4. **Crespo Ramirez, Rigel y Alonso Cardenas, Zoraida.** Inteligencia Empresarial. Qué y Cómo. La Habana, Cuba. : Instituto de Información Científica y Tecnológica, IDICT, 2009. ISBN 978-959-234-070-1.
5. La planificación. **Pedraza Quintero, Julio César.** Málaga : Grupo Eumed.net (Universidad de Málaga), 2009. ISSN 1696-8360.
6. **P. Robbins, Stephen.** Administración. México : Pearson Educación de México, S.A, 2005. ISBN: 970-26-0555-5.
7. **Benavides Pañeda, Javier .** Administración. s.l. : McGraw Hill, 2004.
8. **Gutiérrez Fuenzalida, Fernando Rene, Tapia Zurita y Alexi Alejandro.** Desarrollo de Sistema Web de Asignación Automática de Mallas de Turnos para Protector Security. Chile : s.n., 2013.
9. **Rubio, Emilio .** Cuando el tiempo es oro. Gestión de calendarios de personal. Soluciones para entornos de crisis. [En línea] Julio de 2013. [Citado el: 10 de febrero de 2015.] <http://www.bspreviews.com>.
10. **TecnoHospital.** [En línea] 3 de abril de 2008. [Citado el: 10 de febrero de 2015.] <http://www.tecnohospital.es>.
11. **Torres Aguilera , Grettell y Almeida Oquendo, Humberto .** Gestor Web para el control de la Guardia Obrera de la Universidad de las Ciencias Informáticas (UCI). La Habana, Cuba : s.n., 2009.
12. **Bejerano González, Karel y Montes de Oca Barrios, Félix Miguel .** Aplicación Web para el control de la Guardia Estudiantil y la Cuartelería de la Facultad 7. La Habana, Cuba : s.n., 2009.
13. **Campos León, Arlety y Méndez Méndez, Verónica.** Sistema de gestión docente metodológica integrado con la plataforma informativa SO3. La Habana, Cuba : s.n., 2013.
14. **Jiménez Barreto, Nardely y Concepción Fernández, Yasser.** Aplicación web para la gestión de la planificación de a guardia obrera estudiantil en la Universidad de las Ciencias Informáticas. La Habana, Cuba : s.n.

15. **Mata Sánchez, Dailín y Navarro Quintero, Aroldo** . Sistema para la Gestión de la Guardia Obrera-Estudiantil en la Facultad 3. La Habana, Cuba : s.n., 2013.
16. **Letelier , Patricio y Penadés, M. Carmen.** Universidad Politécnica de Valencia. [En línea] 2005-2011. [Citado el: 3 de febrero de 2015.] <https://www.upv.es> .
18. **Rodríguez Sánchez, Tamara.** Programa de Mejora. Metodología de desarrollo para la Actividad productiva de la UCI. Universidad de las Ciencias Informáticas, La Habana, Cuba : s.n., 2014.
19. **slideshare.** [En línea] 25 de noviembre de 2013. [Citado el: 10 de febrero de 2015.] <http://es.slideshare.net/luchoapazam1/desarrollo-de-aplicaciones-con-rup-y-uml>.
20. **Rosales Morales, Yanet, Marrero Clark, Michael Eduardo y Trujillo Oliva, Adrian.** Extensión de la herramienta Visual Paradigm para la generación de clases de acceso a datos con Doctrine 2.0. No. 10, Universidad de las Ciencias Informáticas, La Habana, Cuba : s.n., 2013, Vol. Vol. 6. ISSN: 2306-2495.
21. **Sanjurjo Badía.** DesarrolloWeb.com . [En línea] APACHEctl. [Citado el: 10 de febrero de 2015.] <http://www.desarrolloweb.com/php/>.
22. **Potencier, Fabien , Weaver, Ryan y Eguiluz, Javier** . Buenas Prácticas para Aplicaciones Symfony.
23. **Sánchez González, Idel Jorge.** Curso Básico de Symfony2. Universidad de las Ciencias Informáticas, La Habana, Cuba : s.n., 2012.
24. **Martinez Guerrero, Rafael.** PostgreSQL-es. [En línea] 23 de marzo de 2009. http://www.postgresql.org.es/sobre_postgresql.
25. **Rosales, Pere y Pamies, Dani.** Inusual. [En línea] enero de 2014. <http://inusual.com/articulos/phpstorm-8-con-soporte-para-drupal-8-y-wordpress/>.
26. Servidores Web – Apache. **Pavón Mestras, Juan.** Madrid : s.n.
27. **Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato.** Control de versiones con Subversion. California, EE.UU : s.n., 2004.
28. PortalProgramas. [En línea] 2003. [Citado el: 10 de febrero de 2015.] <http://www.portalprogramas.com/rapidsvn/>.
29. **Pressman, R.** Ingeniería de software. Un enfoque práctico. 6ta Edición. 2007.
30. **Sommerville, Ian.** Ingeniería de software. Séptima edición. Madrid. España : Pearson Educación S.A, 2005. ISBN: 84-7829-074-5.

31. **Rodríguez, Daniel.** Análisis y Diseño.
32. **Barrera León, Luisa Fernanda .** Pontificia Universidad Javeriana. [En línea] 2011. [Citado el: 15 de 3 de 2015.] http://pegasus.javeriana.edu.co/CIS0930IS08DocumentosSAD_ZuGym.docx.
33. **yiiframework.** [En línea] 3 de 12 de 2008. [Citado el: 5 de 3 de 2015.] <http://www.yiiframework.com/doc/guide/1.1/en/basics.best-practices>.
34. **Eguiluz, Javier.** Desarrollo Web Ágil Con Symfony2.
35. **Larman, Craig.** UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado . s.l. : Prentice Hall.
36. **Pressman , Roger.** Ingeniería del Software. Un enfoque práctico. Séptima Edición. s.l. : MCGRAW-HILL, 2010. ISBN 9786071503145.
37. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** El Lenguaje Unificado de Modelado. Manual de Referencia. s.l. : Addison Wesley.
38. **Microsoft. Developer Network.** [En línea] 2007. [Citado el: 10 de 3 de 2015.] <https://msdn.microsoft.com/es-es/library>.
39. **Rodríguez Tello, Dr. Eduardo A.** Cinvestav Tampulinas. [En línea] 7 de 2006. [Citado el: 2 de 5 de 2015.] <http://www.tamps.cinvestav.mx/~ertello/swe/sesion15.pdf>.
40. **Universidad Autónoma de Baja California.** Facultad de Ciencias Químicas e Ingeniería. [En línea] febrero de 2005. [Citado el: 2 de 5 de 2015.] fcqi.tij.uabc.mx/usuarios/luisgmo/data/8.3%20prb-cal-mant.pdf.
41. **Blanco Bueno, Carlos.** Universidad de Cantabria. [En línea] 9 de 12 de 2013. [Citado el: 2 de 5 de 2015.] <http://ocw.unican.es/enseanzas-tecnicas/ingenieria-del-software-ii>. ISBN: 978-84-694-5304-9.
42. **Kusmina, N. V.** Metódicas investigativas de la actividad pedagógica. In.: Leningrado, 1970.

ANEXOS

Anexo 1: Diccionario de datos de la entidad Área

Descripción	Es la entidad que representa el área donde se realizará la GOE en la facultad 3, por ejemplo si es en el docente, residencia, etc.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
nombre	Nombre del área donde se realizará la GOE en la facultad 3, ya sea docente, residencia, etc.	Caracteres alfabéticos	No	No	Caracteres alfabéticos	Caracteres especiales, Números

Anexo 2: Diccionario de datos de la entidad Categoría

Descripción	Es la entidad que representa la categoría que puede tener una persona, por ejemplo si es estudiante, trabajador, etc.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
nombre	Categoría que recibe una persona	Caracteres alfabéticos	No	No	Caracteres alfabéticos	Caracteres especiales, Números
descripción	Breve descripción del tipo de categoría	Caracteres alfabéticos	Si	No	Caracteres alfabéticos	Caracteres especiales, Números

Anexo 3: Diccionario de datos de la entidad Guardia

Descripción	Es la entidad que representa el nombre de la GOE que se va a realizar y donde se realizará.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
nombre	Nombre que se le quiera poner a la GOE.	Caracteres alfabéticos	No	No	Caracteres alfabéticos	Caracteres especiales, Números

Anexo 4: Diccionario de datos de la entidad Incidencia

Descripción	Es la entidad que representa las incidencias ocurridas en la GOE, como por ejemplo motivos de ausencia a la GOE por parte de los estudiantes, trabajadores, etc.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas

descripción	Describe las breves causas de las incidencias aparecidas en el transcurso de la guardia.	Carácteres alfabéticos	Si	No	Carácteres alfabéticos	Carácteres especiales, Números
rango de fecha	Representa el rango de fecha para buscar incidencias ocurridas en esa fecha	Carácteres especiales, Números	No	No	Carácteres especiales, Números	Carácteres alfabéticos

Anexo 5: Diccionario de datos de la entidad Integrante

Descripción	Es la entidad que representa a cada integrante involucrado en la GOE.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
función	Función del involucrado en la GOE, por ejemplo si es Jefe de Turno, etc.	Caracteres alfabéticos	No	Si	Carácteres alfabéticos	Carácteres especiales, Números
sexo	Sexo del integrante del equipo de GOE.	Caracteres alfabéticos	No	No	Carácteres alfabéticos	Carácteres especiales, Números

Anexo 6: Diccionario de datos de la entidad Observación

Descripción	Es la entidad que representa todas las observaciones realizadas a los trabajadores y estudiantes involucrado con la GOE.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
descripción	Describe las restricciones que puedan tener los trabajadores, es decir, motivos por los cuales no puedan hacer la GOE.	Caracteres alfabéticos	Si	No	Carácteres alfabéticos	Carácteres especiales, Números
días no válidos	Si el trabajador tiene alguna observación, el atributo describe los días en los cuales no puede realizar la GOE.	Caracteres alfabéticos	Si	No	Carácteres alfabéticos	Carácteres especiales, Números

Anexo 7: Diccionario de datos de la entidad Permuta

Descripción	Es la entidad que representa los cambios que puede realizar una persona en la planificación correspondiente a su GOE.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas

Fecha	Representa la fecha en que se realiza la permuta.	Números	No	Si	Números	Carácteres especiales, Carácteres alfabéticos
estado	Describe si la permuta fue aceptada, rechazada, cancelada, etc.	Carácteres alfabéticos	No	No	Carácteres alfabéticos	Números

Anexo 8: Diccionario de datos de la entidad Persona

Descripción		Describe todos los datos necesarios de la persona involucrada con la GOE.				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
nombre	Nombre de la persona.	Caracteres alfabéticos	No	No	Carácteres alfabéticos	Carácteres especiales, Números
apellidos	Apellidos de la persona.	Caracteres alfabéticos	No	No	Carácteres alfabéticos	Carácteres especiales, Números
sexo	Sexo de la persona.	Caracteres alfabéticos	No	No	Carácteres alfabéticos	Carácteres especiales, Números
correo	Correo de la persona.	Caracteres alfabéticos	No	No	Caracteres alfabéticos	Carácteres especiales, Números
usuario	Usuario de la persona por si es necesario enviar alguna información por correo electrónico.	Caracteres alfabéticos	No	Si	Caracteres alfabéticos	Carácteres especiales, Números
área	Área a la que pertenece el involucrado en la GOE. Ejemplo: si es un trabajador, si pertenece a un proyecto, etc. Si es un estudiante, brigada a la que pertenece, etc.	Números, Caracteres alfabéticos	No	No	Números, Caracteres alfabéticos	Carácteres especiales

Anexo 9: Diccionario de datos de la entidad Planificación

Descripción		Representa la planificación de la GOE, es decir, recoge todos los datos necesarios para poder ejecutar la GOE por parte de los trabajadores y estudiantes.				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
fecha	Rango de fecha para realizar la GOE.	Numérico	No	No	Números	Carácteres especiales,

						Caracteres alfabéticos
cumplimiento	Recoge si cada persona que realizó la GOE la cumplió o no.	Caracteres alfabéticos	No	No	Caracteres alfabéticos	Caracteres especiales, números
motivo incumplimiento	Recoge el motivo de cada persona que no realizó la GOE.	Caracteres alfabéticos	Si	No	Caracteres alfabéticos	Caracteres especiales, números

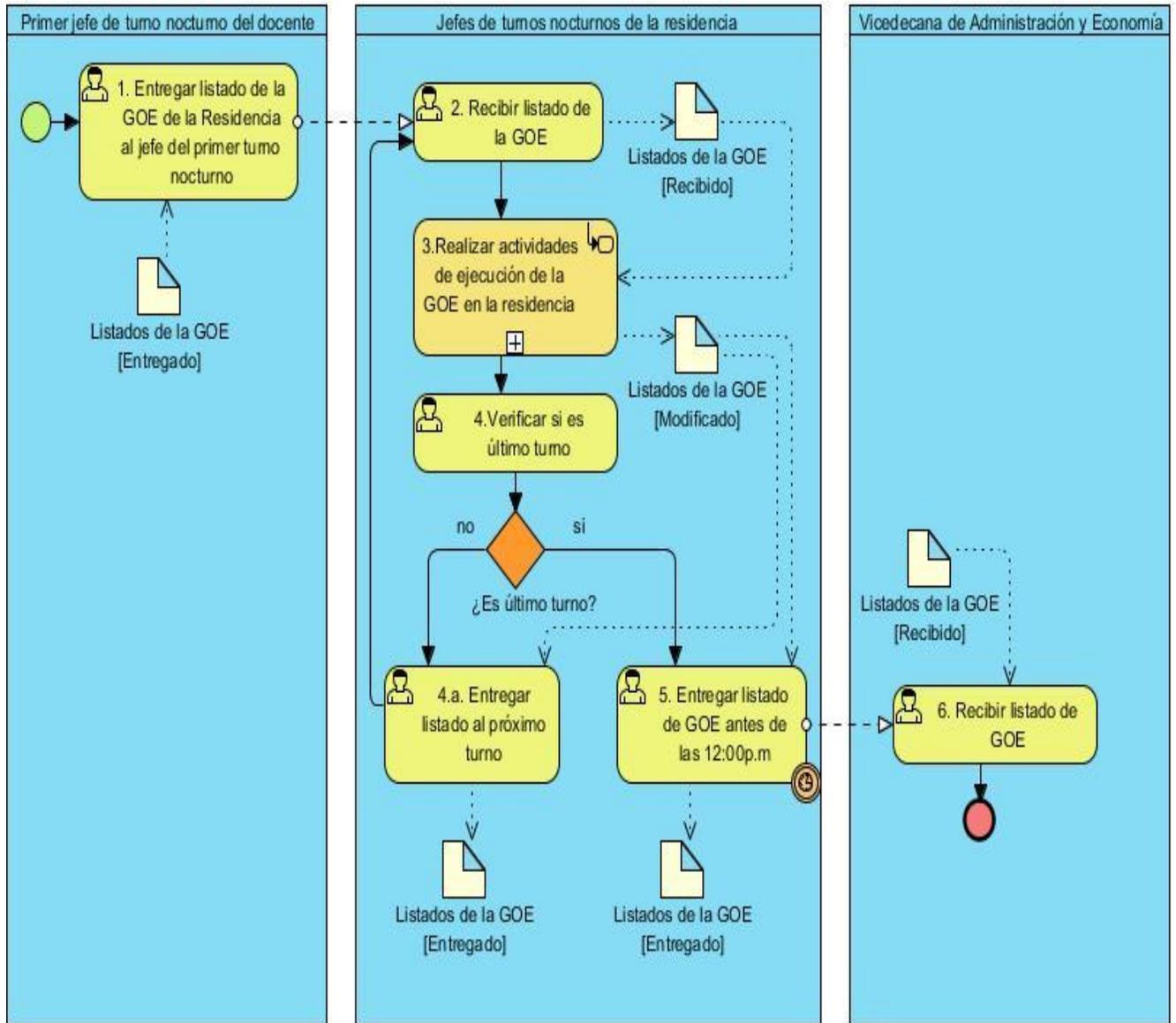
Anexo 10: Diccionario de datos de la entidad Posta

Descripción	Es la entidad que representa la localización del lugar donde estará ubicado el equipo de GOE con sus integrantes.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
número de posta	Describe el número de la posta en el que a las partes involucradas les toca realizar la GOE.	Números	Si	No	Números	Caracteres especiales, Caracteres alfabéticos
Localización	Describe la localización de la posta.	Números, Caracteres alfabéticos	Si	No	Números Caracteres alfabéticos	Caracteres especiales

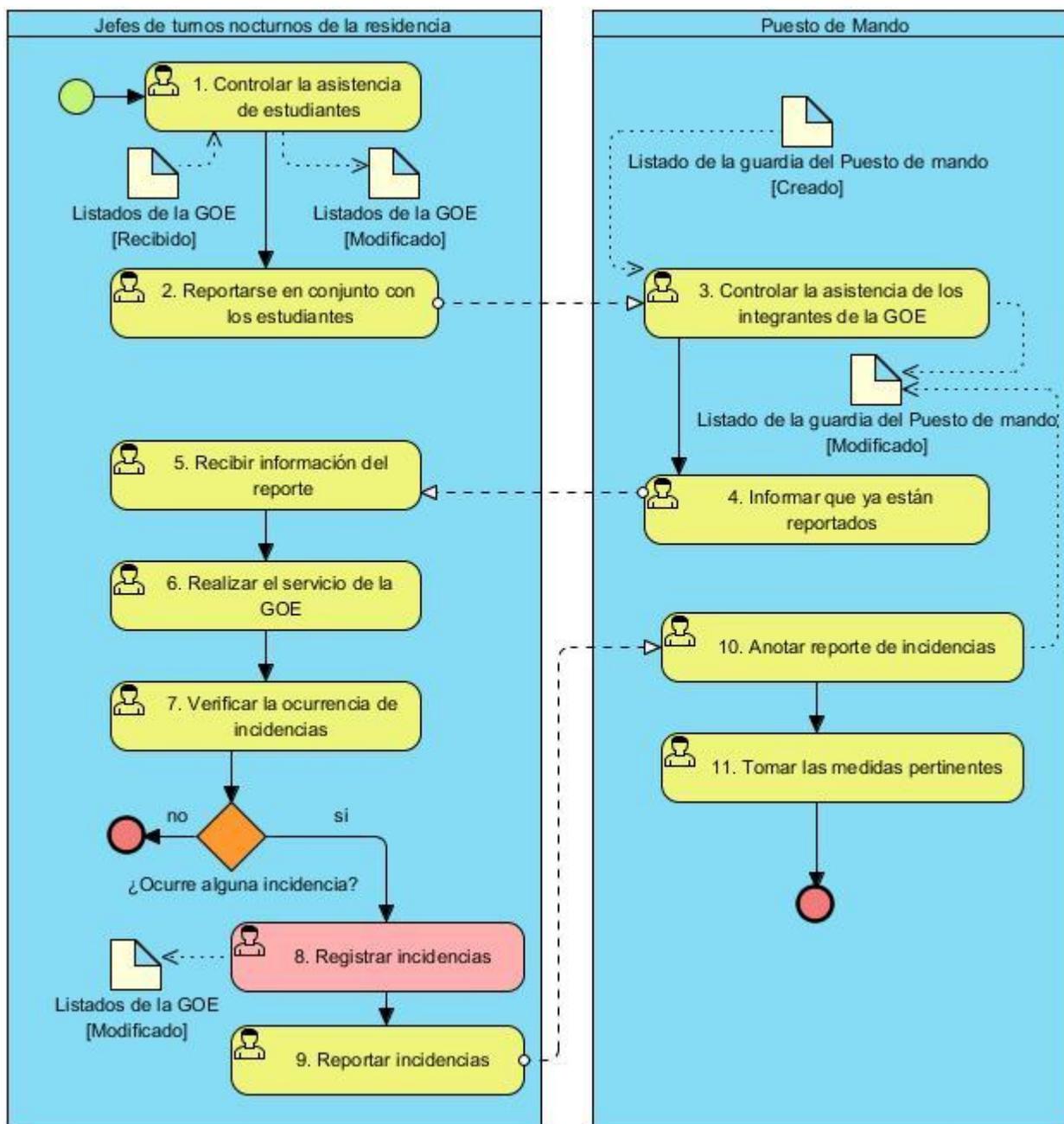
Anexo 11: Diccionario de datos de la entidad Turno

Descripción	Describe información del turno de guardia que le toca realizar a los involucrados en la GOE.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
hora de inicio	Hora de inicio del turno de la GOE.	Números	No	No	Números	Caracteres especiales, Caracteres alfabéticos
hora de fin	Hora de fin del turno de la GOE.	Números	No	No	Números	Caracteres especiales, Caracteres alfabéticos

Anexo 12: Diagrama del proceso de negocio Ejecutar GOE en la Residencia



Anexo 13: Diagrama del subproceso de negocio Realizar actividades de ejecución de la GOE en la residencia

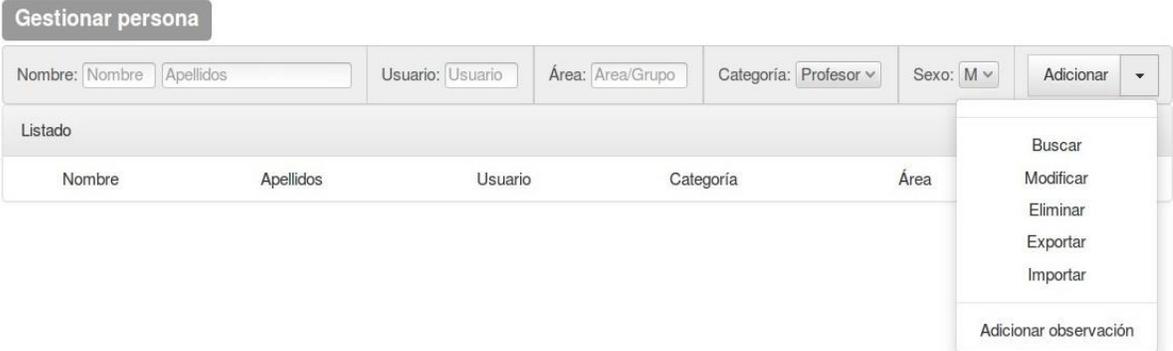


Anexo 14: Descripción del proceso Ejecutar GOE en la Residencia

Objetivo	Realizar la GOE en la residencia.
Evento(s) que lo genera(n)	Entregar listado de la GOE de la Residencia al jefe del primer turno nocturno.
Pre condiciones	Debe existir el Excel con la planificación de la GOE.
Marco legal	N/A.
Reglas de negocio	N/A

Responsable	Jefes de turnos nocturnos de la residencia.
Clientes internos	N/A.
Clientes externos	N/A.
Entradas	N/A.
Flujo de eventos	
Flujo básico Ejecutar GOE en la Residencia.	
1.	Entregar listado de la GOE de la residencia al jefe del primer turno nocturno: El jefe del primer turno nocturno del docente le entrega el listado de la GOE al jefe del primer turno nocturno de la residencia.
2.	Recoger listado de la GOE: El jefe del primer turno nocturno recoge el listado de la GOE.
3.	Realizar actividades de ejecución de la GOE en la residencia: En caso de que el turno de guardia sea el último, el jefe de turno realiza las actividades comunes para todos los turnos de guardia nocturnos. Ver imagen de subproceso Realizar actividades de ejecución de la GOE en la residencia.
4.	Verificar si es último turno: El jefe del turno verifica que su turno no sea el último.
5.	Entregar listado de la GOE antes de las 12:00 p.m.: El jefe del último turno nocturno entrega el listado de la GOE a la Vicedecana de Economía y Administración antes de las 12:00 p.m.
6.	Recoger listado de la GOE: La Vicedecana de Economía y Administración recoge el libro de la GOE.
7.	Concluye el proceso.
Pos-condiciones	
1.	Se ejecutó la GOE en la residencia.
Salidas	
1.	N/A
Flujos paralelos	
N/A	
Pos-condiciones	
1.	N/A
Salidas	
1.	N/A
Flujos alternos	
1.	Flujo alternativo de la actividad 3. Verificar si es último turno
	4. a. Entregar listado de la GOE al próximo turno.
	4. b. Ir a subproceso Realizar actividades de ejecución de la GOE en la residencia.
Pos-condiciones	
1.	N/A
Salidas	
1.	N/A
Asuntos pendientes	
N/A	

Anexo 15: Descripción del Gestionar persona de la GOE el RF Adicionar persona

Número: RF_30		Nombre del requisito: Adicionar persona	
Programador:		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 7	
Riesgo en Desarrollo: Alto		Tiempo Real: Una semana	
Descripción: Adiciona a todas las personas que sus facultades físicas le permitan realizar la GOE mediante una búsqueda en la red por su usuario o exportando un Excel que contenga el listado de trabajadores y estudiantes de la Facultad 3.			
Observaciones:			
Prototipo de interfaz:			
 <p>The screenshot shows a web interface titled "Gestionar persona". It features a search bar with fields for "Nombre" (with sub-fields "Nombre" and "Apellidos"), "Usuario", "Área" (with sub-field "Área/Grupo"), "Categoría" (set to "Profesor"), and "Sexo" (set to "M"). There is an "Adicionar" button with a dropdown menu. The dropdown menu is open, showing options: "Buscar", "Modificar", "Eliminar", "Exportar", "Importar", and "Adicionar observación". Below the search bar is a "Listado" table with columns for "Nombre", "Apellidos", "Usuario", "Categoría", and "Área".</p>			

Anexo 16: Descripción del Gestionar observación el RF Adicionar observación

Número: RF_35		Nombre del requisito: Adicionar observación	
Programador:		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 2	
Riesgo en Desarrollo: Medio		Tiempo Real:	
Descripción: Comienza cuando se realiza la búsqueda del usuario en la red, luego de hallarlo se le agregan los días en que no pueda realizar la GOE y una breve descripción de la restricción del mismo.			
Observaciones:			

Prototipo de interfaz:

Adicionar observación x

Nombre: Oneisy

Días no válidos: Lunes ▾

Turnos no válidos: 3:00am ▾

Descripción: Embarazada

Cerrar
Adicionar

Anexo 17: Descripción del Gestionar GOE el RF Adicionar GOE

Número: RF_21	Nombre del requisito: Adicionar GOE				
Programador:	Iteración Asignada: 1				
Prioridad: Alta	Tiempo Estimado: 2				
Riesgo en Desarrollo: Medio	Tiempo Real:				
<p>Descripción: Comienza cuando la Vicedecana de Economía y Administración decide adicionar la GOE, seleccionando el nombre que desee y el área donde se realizará la GOE, ya sea en el docente, en la residencia o en cualquier otra área que estime conveniente.</p>					
Observaciones:					
<p>Prototipo de interfaz:</p> <div style="border: 1px solid gray; padding: 10px;"> <p style="background-color: #d9d9d9; padding: 5px; margin-bottom: 10px;">Gestionar GOE</p> <p>Nombre: <input style="width: 150px;" type="text"/> Área: Docente ▾ Adicionar ▾</p> <p style="background-color: #d9d9d9; padding: 5px; margin-bottom: 10px;">Guardias creadas</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #d9d9d9;"> <th style="width: 60%; padding: 5px;">Nombre</th> <th style="width: 40%; padding: 5px;">Á</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;"></td> <td style="padding: 5px; text-align: right;"> Eliminar Modificar </td> </tr> </tbody> </table> </div>		Nombre	Á		Eliminar Modificar
Nombre	Á				
	Eliminar Modificar				

Anexo 18: Descripción del Gestionar postas de la GOE el RF Adicionar posta

Número: RF_13	Nombre del requisito: Adicionar posta
Programador:	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 2

Riesgo en Desarrollo: Bajo	Tiempo Real:				
Descripción: Comienza cuando la Vicedecana de Economía y Administración decide adicionar postas a los turnos de la guardia poniéndole el número de la misma y su localización, es decir, donde se encuentran ubicadas las postas.					
Observaciones:					
Prototipo de interfaz:					
<div style="border: 1px solid #ccc; padding: 10px;"> <div style="background-color: #808080; color: white; padding: 5px; text-align: center; font-weight: bold; margin-bottom: 10px;">Gestionar postas</div> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 30%;"> Número: <input type="text" value="Número"/> </div> <div style="width: 30%;"> Localización: <input type="text" value="Localización"/> </div> <div style="width: 20%; text-align: right;"> <input type="button" value="Adicionar"/> </div> </div> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> Postas existentes </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 50%;">Número</th> <th style="width: 50%;">Localización</th> </tr> </thead> <tbody> <tr> <td style="height: 20px;"> </td> <td> </td> </tr> </tbody> </table> </div>		Número	Localización		
Número	Localización				

Anexo 19: Descripción del Gestionar turnos de la GOE el RF Adicionar turno

Número: RF_10	Nombre del requisito: Adicionar turno				
Programador:	Iteración Asignada: 1				
Prioridad: Alta	Tiempo Estimado: 2				
Riesgo en Desarrollo: Medio	Tiempo Real:				
Descripción: Comienza cuando la Vicedecana de Economía y Administración decide adicionar turnos a la guardia que se escoja, poniéndole la hora en que comienza el turno y la hora en que se acaba					
Observaciones:					
Prototipo de interfaz:					
<div style="border: 1px solid #ccc; padding: 10px;"> <div style="background-color: #808080; color: white; padding: 5px; text-align: center; font-weight: bold; margin-bottom: 10px;">Gestionar turnos</div> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 15%;"> Guardia: <input type="text" value="Guardia X"/> </div> <div style="width: 30%;"> Hora de inicio: <input type="text" value="Hora"/> <input type="text" value="Min"/> <input type="text" value="am"/> </div> <div style="width: 30%;"> Hora de fin: <input type="text" value="Hora"/> <input type="text" value="Min"/> <input type="text" value="am"/> </div> <div style="width: 15%; text-align: right;"> <input type="button" value="Adicionar"/> </div> </div> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> Turnos de guardia </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 50%;">Inicio</th> <th style="width: 50%;">Fin</th> </tr> </thead> <tbody> <tr> <td style="height: 20px;"> </td> <td> </td> </tr> </tbody> </table> <div style="position: absolute; right: 10px; top: 10px; background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <input type="button" value="Eliminar"/> <input type="button" value="Modificar"/> </div> </div>		Inicio	Fin		
Inicio	Fin				

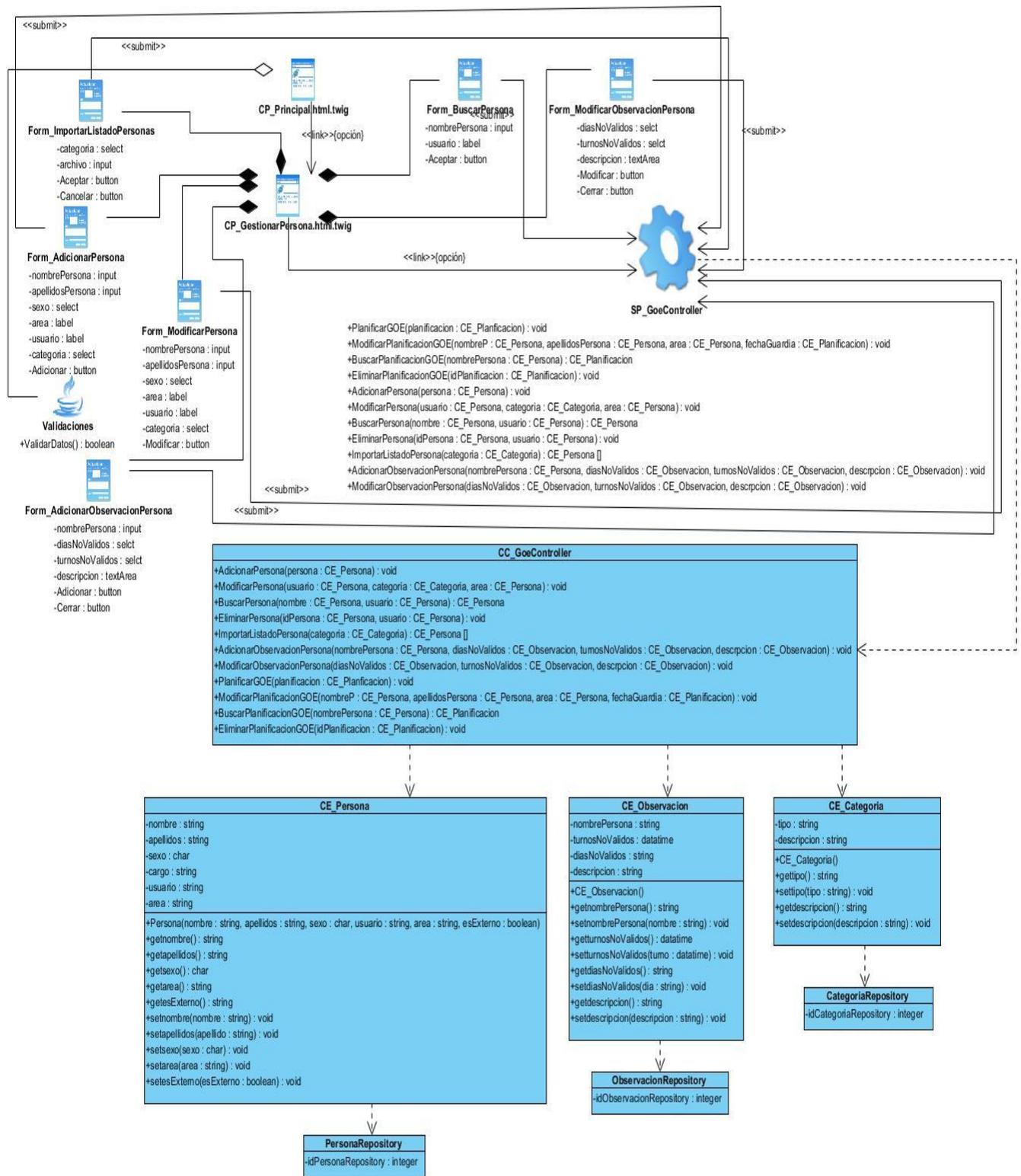
Anexo 20: Descripción del Gestionar equipo por turno el RF Adicionar integrante

Número: RF_27	Nombre del requisito: Adicionar integrante		
Programador:	Iteración Asignada: 1		
Prioridad: Alta	Tiempo Estimado: 5		
Riesgo en Desarrollo: Medio	Tiempo Real:		
Descripción: Comienza cuando la Vicedecana de Economía y Administración decide adicionar integrantes a la GOE que se seleccione, escogiendo el turno al que se integrará, la categoría a la que pertenece, la función que realiza en la GOE y su sexo.			
Observaciones:			
Prototipo de interfaz:			

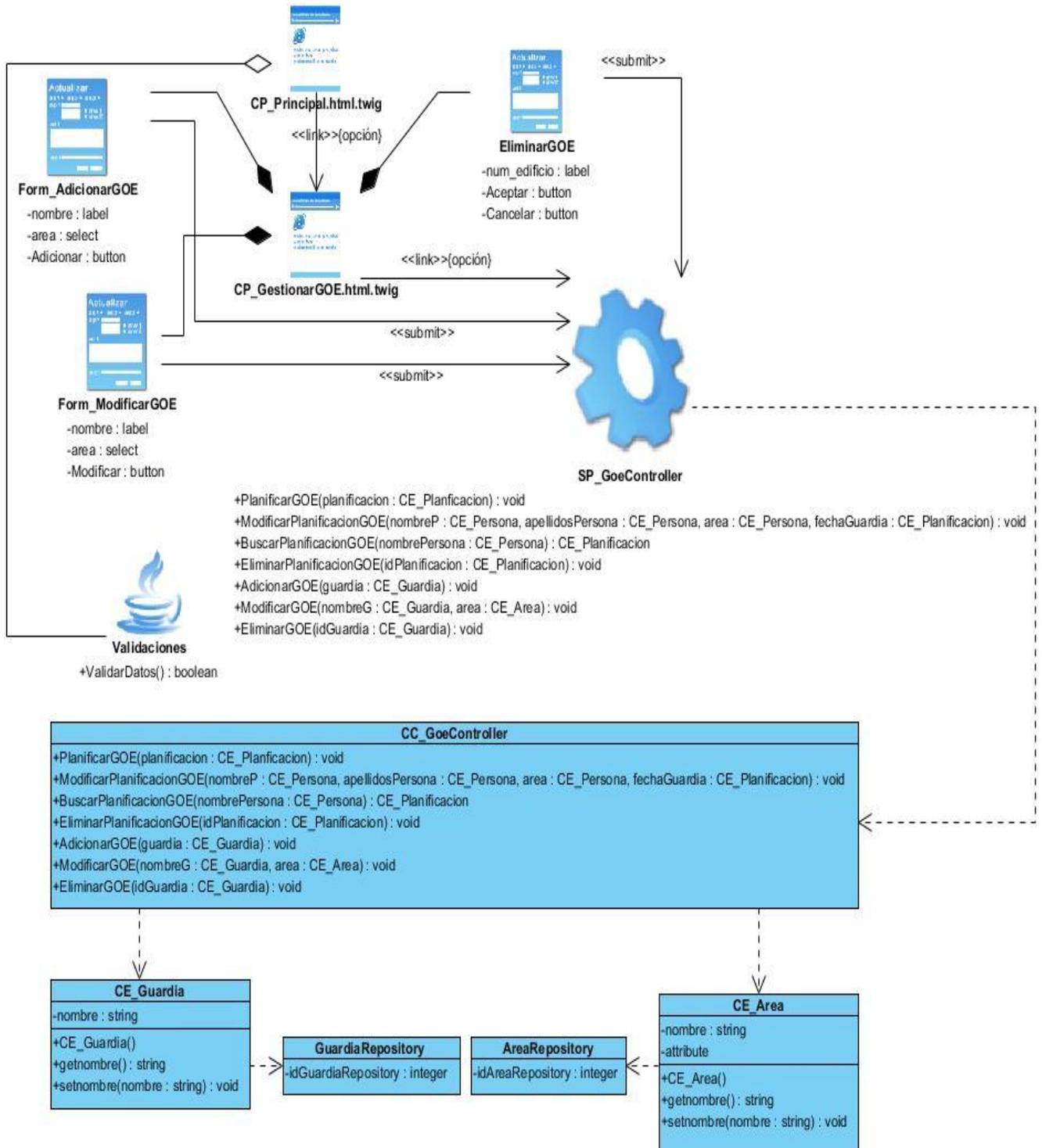
Anexo 21: Descripción del Gestionar categoría el RF Adicionar categoría

Número: RF_40	Nombre del requisito: Adicionar categoría		
Programador:	Iteración Asignada: 1		
Prioridad: Alta	Tiempo Estimado: 2		
Riesgo en Desarrollo: Medio	Tiempo Real:		
Descripción: Comienza cuando la Vicedecana de Economía y Administración decide adicionar la categoría en la que será clasificada una persona involucrada en la GOE, ya sea estudiante, trabajador y una breve descripción de la misma.			
Observaciones:			
Prototipo de interfaz:			

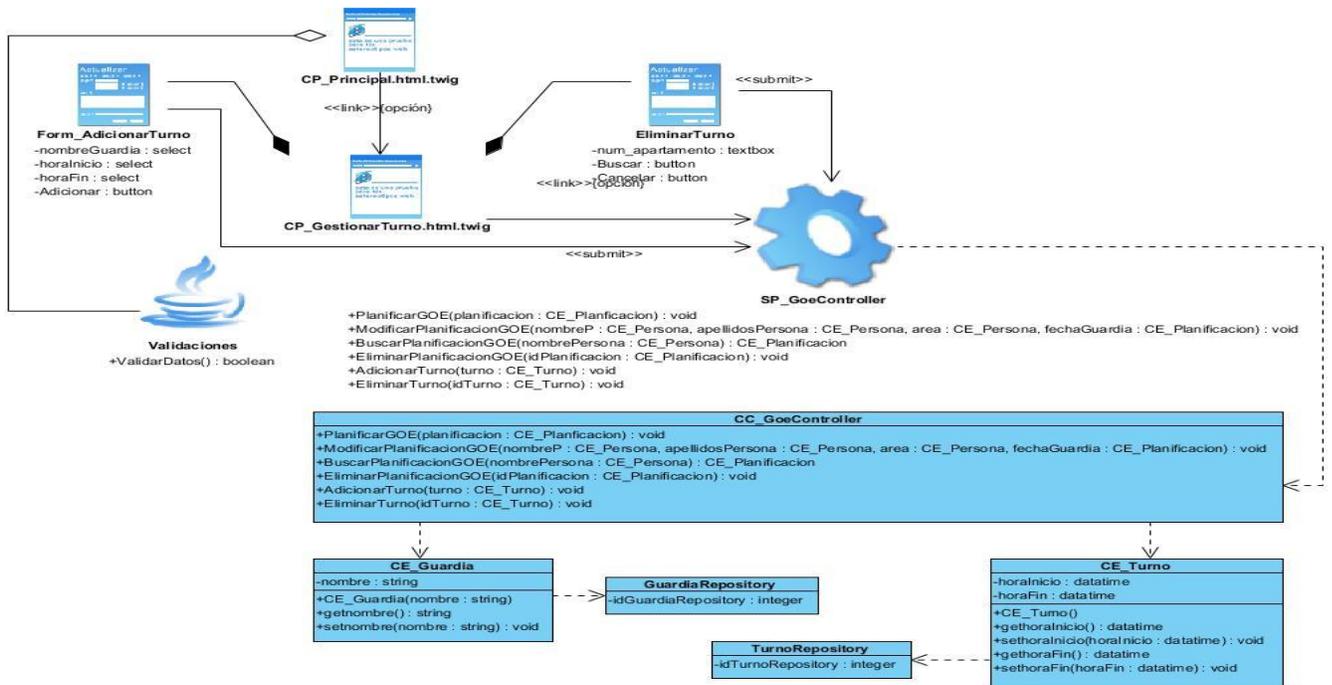
Anexo 22: Diagrama de clases del diseño Gestionar persona



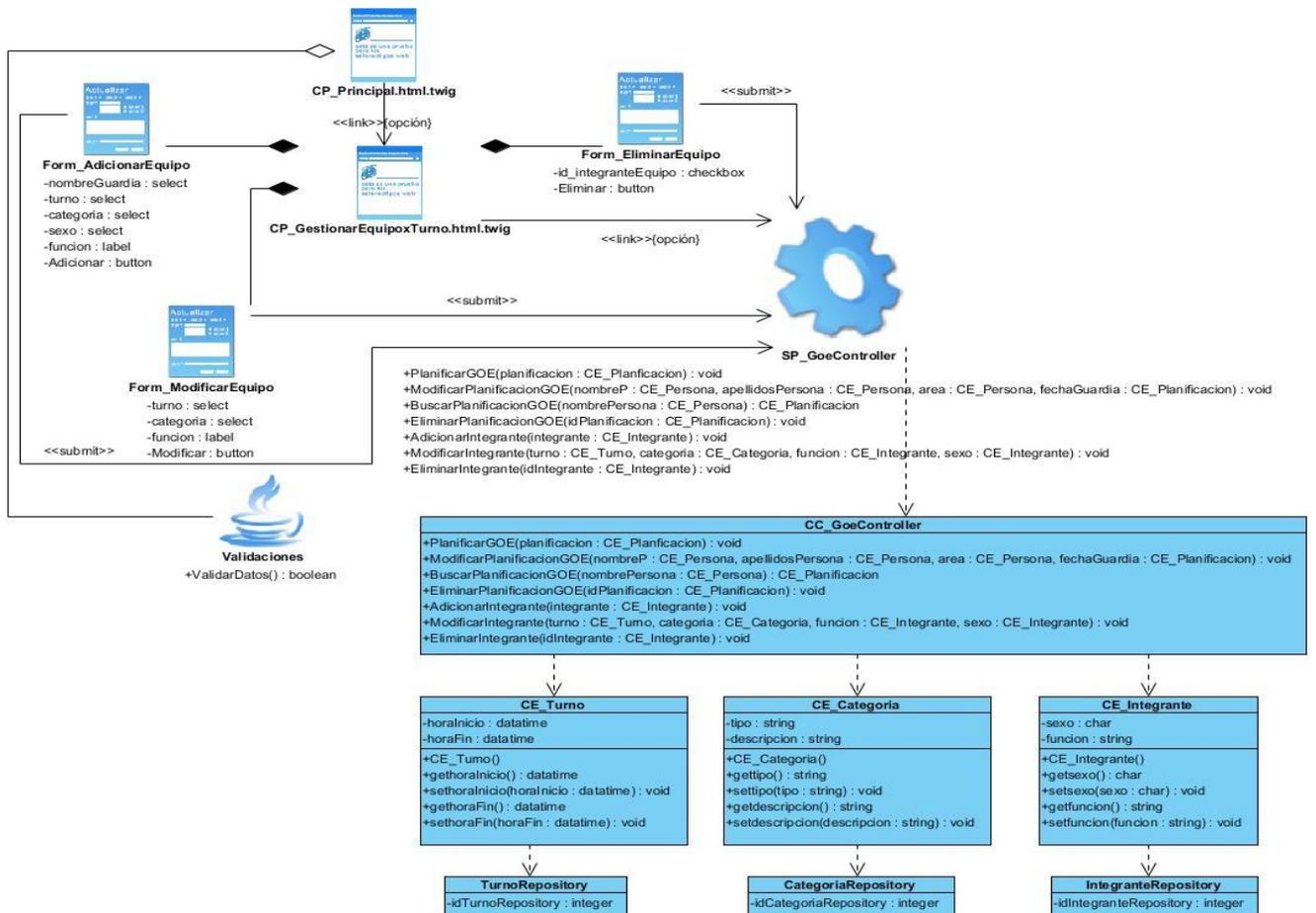
Anexo 23: Diagrama de clases del diseño Gestionar guardia



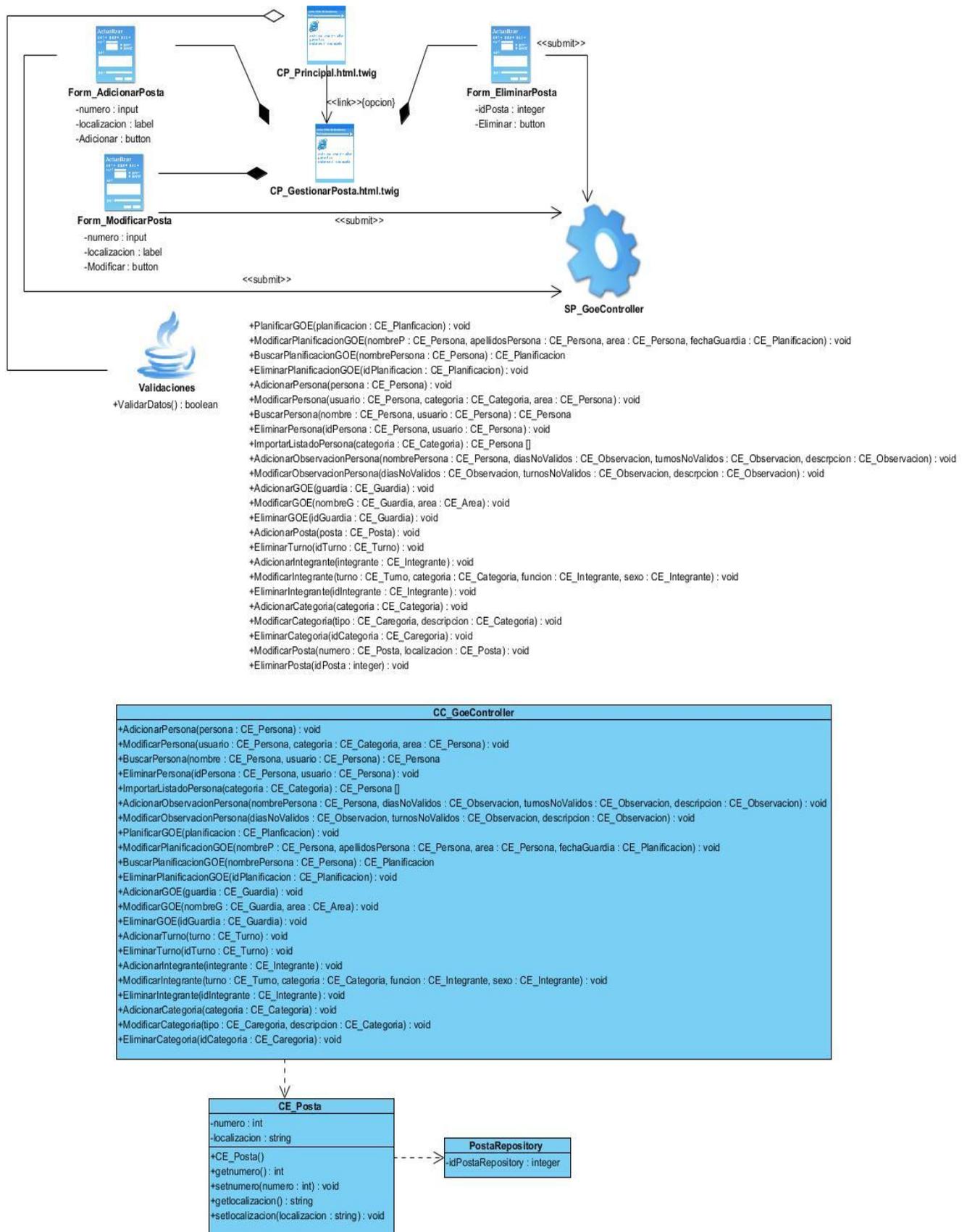
Anexo 24: Diagrama de clases del diseño Gestionar turno



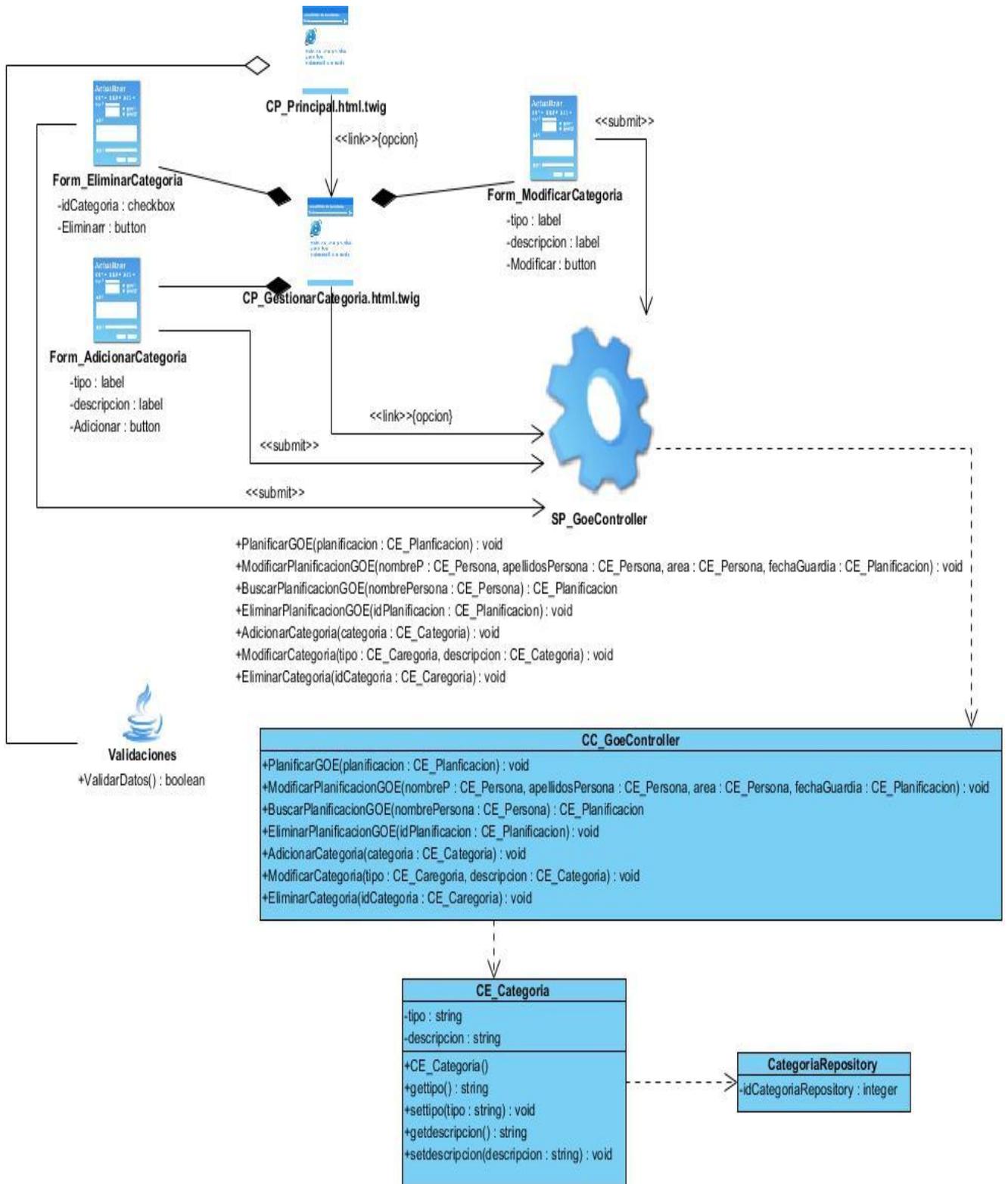
Anexo 35: Diagrama de clases del diseño Gestionar equipo por turno



Anexo 26: Diagrama de clases del diseño Gestionar posta



Anexo 27: Diagrama de clases del diseño Gestionar categoría



Anexo 28: Caso de prueba para el requisito Adicionar Área

Condiciones de ejecución

1- Se debe seleccionar la opción Configuración inicial- Áreas

SC Adicionar Área				
Escenario	Descripción	Nombre	Respuesta del sistema	Flujo central
EC 1.1: Adicionar el Área introduciendo los datos válidos	Se adiciona el área introduciendo los datos válidos.	V Residencia	Se adiciona el área correctamente.	<input type="checkbox"/> Se selecciona la opción Configuración inicial- Áreas. <input type="checkbox"/> Se introducen los datos correctamente. <input type="checkbox"/> Se presiona el botón Adicionar.
EC 1.2: Adicionar el Área dejando campos vacíos.	Se adiciona el área dejando campos vacíos.	/ vacío	El sistema muestra un mensaje de error: "Debe llenar los campos vacíos".	<input type="checkbox"/> Se selecciona la opción Configuración inicial- Áreas. <input type="checkbox"/> Se deja el campo Nombre vacío. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestra un mensaje donde se debe llenar el campo vacío.

Anexo 29: Caso de prueba para el requisito Adicionar GOE

Condiciones de ejecución

1- Se debe seleccionar la opción Configuración inicial- Guardias

2- Debe existir en el sistema al menos un Área.

SC Adicionar Guardias					
Escenario	Descripción	Nombre	Área	Respuesta del sistema	Flujo central
EC 1.1: Adicionar las guardias introduciendo los datos válidos	Se adiciona las guardias introduciendo los datos válidos.	V Guardia Residencia	V Abril Residencia	El sistema adiciona la guardia.	<input type="checkbox"/> Se selecciona la opción Configuración inicial- Guardias. <input type="checkbox"/> Se introducen los datos correctamente. <input type="checkbox"/> Se presiona el botón Adicionar.
EC 1.2: Adicionar las guardias introduciendo los datos inválidos.	Se adiciona las guardias introduciendo los datos inválidos	/ \$-&/%!	/ vacío	El sistema muestra mensajes de error: "Existen datos incorrectos", " Debe existir al menos un área creada"	<input type="checkbox"/> Se selecciona la opción Configuración inicial- Guardias. <input type="checkbox"/> Se introducen los datos incorrectamente. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestran mensajes de error.
EC 1.3: Adicionar las guardias con el campo Nombre vacío y los demás datos correctos.	Se adiciona las guardias con el campo Nombre vacío y los demás datos correctos.	/ vacío	V Residencia	El sistema muestra un mensaje de información: "Debe llenar este campo vacío".	<input type="checkbox"/> Se selecciona la opción Configuración inicial- Guardias. <input type="checkbox"/> Se deja el campo Nombre vacío. <input type="checkbox"/> Se introducen los datos de los restantes campos correctamente. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestra un mensaie de

Anexo 30: Caso de prueba para el requisito Adicionar Postas

Condiciones de ejecución

1- Se debe seleccionar la opción Configuración inicial- Postas

2- Debe existir en el sistema al menos un área.

SC Adicionar Postas						
Escenario	Descripción	Número	Área	Localización	Respuesta del sistema	Flujo central
EC 1.1: Adicionar las postas introduciendo los datos válidos	Se adiciona las postas introduciendo los datos válidos.	1	Residencia	edificio 56	El sistema adiciona las postas.	<input type="checkbox"/> Se selecciona la opción Configuración inicial- Postas. <input type="checkbox"/> Se introducen los datos correctamente. <input type="checkbox"/> Se presiona el botón Adicionar.
EC 1.2: Adicionar las postas introduciendo los datos inválidos.	Se adiciona las postas introduciendo los datos inválidos	uno	vacio	vacio	El sistema muestra mensajes de error: "Existen datos incorrectos". " Debe existir al menos un área creada"y por último un mensaje de información "La posta está sin localización"	<input type="checkbox"/> Se selecciona la opción Configuración inicial- Postas. <input type="checkbox"/> Se introducen los datos incorrectamente. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestran mensajes de error.
EC 1.3: Adicionar las postas con el campo Número vacío y los demás datos correctos.	Se adiciona las postas con el campo Número vacío y los demás datos correctos.	vacio	Residencia	edificio 56	El sistema muestra un mensaje de información: "Debe llenar este campo vacío".	<input type="checkbox"/> Se selecciona la opción Configuración inicial- Postas. <input type="checkbox"/> Se deja el campo Número vacío. <input type="checkbox"/> Se introducen los datos de los restantes campos correctamente. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestra un mensaje de información indicando que no se debe dejar el campo vacío.
EC 1.4: Adicionar las postas con el campo Localización vacío y los demás datos correctos.	Se adiciona las postas con el campo Localización vacío y los demás datos correctos	2	Residencia	vacio	El sistema muestra un mensaje de información: "La posta está sin localización".	<input type="checkbox"/> Se selecciona la opción Configuración inicial- Postas. <input type="checkbox"/> Se introducen los datos de los campos número y área correctamente. <input type="checkbox"/> Se deja el campo localización vacío. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestra un mensaje de información indicando que la posta está sin localización.
EC 1.5: Adicionar las postas con los datos del campo Número repetido y los demás datos correctos.	Se adiciona las postas con los datos del campo Número repetido y los demás datos correctos.	1	Residencia	edificio 56	El sistema muestra un mensaje de error: "Este número de posta ya existe".	<input type="checkbox"/> Se selecciona la opción Configuración inicial- Postas. <input type="checkbox"/> Se introducen los datos de los campos área y localización correctamente y el campo número repetido. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestra un mensaje de error indicando que la posta ya existe.

Anexo 31: Caso de prueba para el requisito Adicionar Categoría

Condiciones de ejecución

1- Se debe seleccionar la opción Gestionar Personas- Categorías

SC Adicionar Categorías					
Escenario	Descripción	Nombre	Descripción	Respuesta del sistema	Flujo central
EC 1.1: Adicionar las categorías introduciendo los datos válidos	Se adicionan las categorías introduciendo los datos válidos.	V Trabajador externo	V Trabajador que no tiene residencia en la Universidad	El sistema adiciona la categoría.	<input type="checkbox"/> Se selecciona la opción Gestionar Personas- Categorías. <input type="checkbox"/> Se introducen los datos correctamente. <input type="checkbox"/> Se presiona el botón Adicionar.
EC 1.2: Adicionar las categorías introduciendo los datos inválidos.	Se adicionan las categorías introduciendo los datos inválidos	/ \$-&%!	N/A vacío	El sistema muestra mensajes de error: "Existen datos incorrectos".	<input type="checkbox"/> Se selecciona la opción Gestionar Personas- Categorías. <input type="checkbox"/> Se introducen los datos incorrectamente. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestran un mensaje de error.
EC 1.3: Adicionar las categorías con el campo Nombre vacío y los demás datos correctos.	Se adicionan las categorías con el campo Nombre vacío y los demás datos correctos.	/ vacío	V Trabajador que no tiene residencia en la Universidad	El sistema muestra un mensaje de información: "Debe llenar este campo vacío".	<input type="checkbox"/> Se selecciona la opción Gestionar Personas- Categorías. <input type="checkbox"/> Se deja el campo Nombre vacío. <input type="checkbox"/> Se introducen los datos de los restantes campos correctamente. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestra un mensaje de información indicando que no se debe dejar el campo vacío.

Anexo 32: Caso de prueba para el requisito Adicionar Turnos

Condiciones de ejecución

1- Se debe seleccionar la opción Turnos

2- Debe existir en el sistema al menos una guardia

SC Adicionar Turnos						
Escenario	Descripción	Guardia	Hora de inicio	Hora de fin	Respuesta del sistema	Flujo central
EC 1.1: Adicionar los turnos introduciendo los datos válidos	Se adicionan los turnos introduciendo los datos válidos.	V Guardia Mayo	V 23:00	V 3:00 AM	El sistema adiciona los turnos.	<input type="checkbox"/> Se selecciona la opción Turnos. <input type="checkbox"/> Se introducen los datos correctamente. <input type="checkbox"/> Se presiona el botón Adicionar.
EC 1.2: Adicionar los turnos introduciendo los datos inválidos.	Se adicionan los turnos introduciendo los datos inválidos	/ vacío	/ vacío	/ vacío	El sistema muestra mensajes de error: No hay ninguna guardia creada en esta área". "Los campos Hora de inicio y Hora de fin no deben estar vacíos".	<input type="checkbox"/> Se selecciona la opción Turnos. <input type="checkbox"/> Se introducen los datos incorrectamente. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestran mensajes de error.
EC 1.3: Adicionar los turnos con el campo Guardia vacío y los demás datos correctos.	Se adicionan los turnos con el campo Guardia vacío y los demás datos correctos.	/ vacío	V 23:00	V 3:00	El sistema muestra un mensaje de información: No hay ninguna guardia creada en esta área".	<input type="checkbox"/> Se selecciona la opción Turnos. <input type="checkbox"/> Se introducen los datos de la hora de inicio y fin del turno correctamente. <input type="checkbox"/> No se ha creado ninguna guardia. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestra un mensaje de información indicando que no hay ninguna guardia creada, por lo que hay que crearla primero para poder gestionar los turnos.

EC 1.4: Planificar la GOE con la configuración de la fecha mal y los demás datos correctos.	Se planifica la GOE con la configuración de la fecha mal y los demás datos correctos.	V	/	/	El sistema muestra un mensaje de error: "Los campos Hora de inicio y Hora de fin no deben estar vacíos".	<input type="checkbox"/> Se selecciona la opción Turnos. <input type="checkbox"/> Se introducen los datos de la guardia correctamente. <input type="checkbox"/> Se introducen los datos de los restantes campos incorrectamente. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestra un mensaje de error indicando que los campos Hora de inicio y Hora de fin no
---	---	---	---	---	--	--

Anexo 33: Caso de prueba para el requisito Adicionar IntegrantexTurno

Condiciones de ejecución

1- Se debe seleccionar la opción Integrantes

2- Debe existir en el sistema al menos una guardia, un turno, una posta y una categoría.

SC Adicionar Integrantes por turno									
Escenario	Descripción	Guardia	Turno	Sexo	Función	Categorías	Postas	Respuesta del sistema	Flujo central
EC 1.1: Adicionar los integrantes introduciendo los datos válidos	Se adicionan los integrantes introduciendo los datos válidos.	V Guardia Mayo	V 23:00-3:00	V F	V Jefe de turno	V Trabajador interno	V primera posta	El sistema adiciona los integrantes por turnos correctamente.	<input type="checkbox"/> Se selecciona la opción Integrantes. <input type="checkbox"/> Se introducen los datos correctamente. <input type="checkbox"/> Se presiona el botón Adicionar.
EC 1.2: Adicionar los turnos introduciendo los datos inválidos.	Se adicionan los turnos introduciendo los datos inválidos	/	/	V F	V Jefe de turno	V Trabajador interno	V primera posta	El sistema muestra mensajes de error: No hay ninguna guardia creada en esta área y "Debe existir un turno creado".	<input type="checkbox"/> Se selecciona la opción Integrantes. <input type="checkbox"/> Se introducen los datos de los campos Guardia y Turno incorrectamente y los demás datos correctamente. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestran mensajes de error.
EC 1.3: Adicionar los turnos con el campo Guardia vacío y los demás datos correctos.	Se adicionan los turnos con el campo Guardia vacío y los demás datos correctos.	V Guardia Mayo	V 23:00-3:00	V F	/ vacío	V Trabajador interno	V primera posta	El sistema muestra un mensaje de información: Por favor chequear este campo que no debe estar vacío".	<input type="checkbox"/> Se selecciona la opción Integrantes. <input type="checkbox"/> Se deja vacío el campo Función y se introducen los demás datos correctamente. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestra un mensaje de información indicando que el campo Función no debe estar vacío.
EC 1.4: Planificar la GOE con la configuración de la fecha mal y los demás datos correctos.	Se planifica la GOE con la configuración de la fecha mal y los demás datos correctos.	V Guardia Mayo	V 23:00-3:00	V F	V Jefe de turno	/ vacío	/ vacío	El sistema muestra un mensaje de error: "Debe existir al menos una categoría y una posta que seleccionar".	<input type="checkbox"/> Se selecciona la opción Integrantes. <input type="checkbox"/> Se introducen los demás datos correctamente. <input type="checkbox"/> Se dejan los campos Categorías y Postas vacíos. <input type="checkbox"/> Se presiona el botón Adicionar. <input type="checkbox"/> Se muestran mensajes de error indicando que los campos Categorías y Postas no deben estar vacíos.

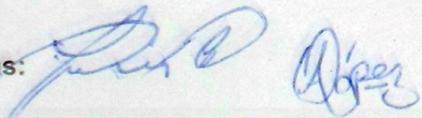
Anexo 34: Acta de aceptación del módulo GOE.

ACTA DE ACEPTACIÓN

En cumplimiento del **Convenio de colaboración en la Universidad de las Ciencias Informáticas** y en función de la ejecución del trabajo de diploma: Módulo GOE para el Sistema de Administración y Economía de la Facultad 3, se hace entrega de los productos que se relacionan a continuación:

- ◆ Expediente de proyecto generado según la metodología.
- ◆ Código de la aplicación y Backup de la Base de Datos.

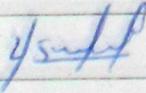
La Parte Cliente, luego de haber revisado los productos de trabajo determina que se **sí se aceptan**.

Entrega	Recibe
Nombres y apellidos: Oneisy Aguilar López Luis Angel Castillo Rodriguez	Nombre y apellidos: MSc. Ana Marys Garcia Rodriguez
Cargos: Diplomantes	Cargo: Vicedecana de Economía y Administración de la Facultad 3
Firmas: 	Firma: 

Observador independiente**Nombre y Apellidos:**

Arnolis Salgueiro Arzuaga

Yunet Suárez Abrante

Cargos: Tutores**Firmas:** 

Fecha: 08/06/2015