

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.**



**FACULTAD 4**

**Trabajo de Diploma**

**Título:**

***Sistema de planificación y control de la Guardia Obrera-Estudiantil de la Facultad 4.***

**Autores: Sifredo Osvaldo Escalona Ramos**

**Wilson Alejandro Ojeda Grass**

**Tutor(es): Lic. Antonio Gutiérrez Laborit**

**Ing. Susana Vidal Cabezas**

**La Habana, junio de 2015  
“Año 56 de la Revolución”**

**Declaración de autoría**

Declaramos que somos los únicos autores del trabajo "Sistema de planificación y control de la Guardia Obrera Estudiantil de la Facultad 4" y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Autores:**

\_\_\_\_\_

Sifredo Osvaldo Escalona Ramos

\_\_\_\_\_

Wilson Alejandro Ojeda Grass

**Tutores:**

\_\_\_\_\_

Lic. Antonio Gutiérrez Laborit

\_\_\_\_\_

Ing. Susana Vidal Cabezas



**“Todos tus sueños pueden hacerse realidad si tienes el coraje de perseguirlos”.**

**Walter Elias (Walt Disney)**

## **Dedicatoria**

**Wilson**

A mi papá, mi mamá y mi hermana, por ser las personas más especiales e importantes de mi vida.

**Sifredo**

A mi mamá, mi papá y mi padrastro por todo el apoyo que me han dado siempre. A mi profesora Rosa de 1er grado, por su dedicación y motivación en mis primeros pasos.

## Agradecimientos

**Wilson**

A mi papá, por ser mi ejemplo a seguir en todo momento, por haber confiado en mí y nunca dudar que llegaría a ser ingeniero. Por todo su amor, cariño, apoyo y sacrificio. Por todos los momentos que dedicaste a hablar conmigo y enseñarme el camino a seguir para ser una mejor persona. Te quiero.

A mi mamá por todo el amor que me ha dado siempre, por apoyarme en todos los momentos difíciles de estos 5 años de universidad. Por su sacrificio, preocupación y entrega incondicional. Te amo por ser la mejor mamá del mundo.

A mi hermana linda, a la quiero con todo mi corazón, por darme su infinito cariño. Por ser hermana y mejor amiga a la misma vez. Por toda la confianza y apoyo que me has dado siempre sin dudarlo.

A mis amigos Felix y Osiel por formar parte de mi vida y ser incondicionales. Los considero mis hermanos por estar a mi lado pase lo que pase y darme su apoyo y confianza.

A mi profe Alison, por ayudarme en momentos difíciles y convertirse en mí amigo. Gracias.

A mi compañero de tesis Osvaldo, otro de mis grandes amigos, por ser como es, porque es único. Nunca cambies mi hermano Huskar.

A todos mis compañeros de estudio y amigos de estos 5 años en la UCI. En especial a Yanet, Gabriel, Dariel (La\_Masacre), Kelvis, Maikel, Armando (Itachy), el Leo Martínez, el Juanca y Jorgito (el pina).

A José Luis (Kakashi), Ariel (Bzonga) y Adonis (Zonido), ustedes se convirtieron en mis amigos también.

A mis tutores Susana y Tony, por guiarme en esta importante tarea. Gracias por su apoyo.

A todos los profes de la Facultad 4 que influyeron en mi formación como profesional en los 5 años de la carrera.

A la UCI, por haber vivido en ella momentos que nunca olvidaré.

## **Sifredo**

A mi mamá, mi papá y Manolo por todo el apoyo y el amor que he recibido de su parte. Por todo lo que me han enseñado, gracias.

A mis queridos hermanos Melissa, María de Lourdes, Luis Miguel y Manuel Francisco, por el cariño que me han dado y la confianza depositada en mi como su hermano mayor.

A mis abuelitos Celia, Miguel, Martha y Argelio por su amor incondicional y toda la ternura con que me trataron siempre. Los quiero mucho.

A mi tía Odalis por su ayuda en los 5 años de universidad, por su cariño, apoyo y comprensión.

A Lourdes Sanamé por ser mi segunda madre y cuidarme siempre.

A mi maestra Rosa por todo lo que fue capaz de enseñarme.

A mis amigos de siempre Jorge, Victor, Adrian, Lordchandis, Aynel, Eddy y Luis Ángel.

A mi compañero de tesis Wilson, por la amistad que hemos construido en estos 5 años de estudio y los momentos que compartimos.

A mis compañeros y amigos Leonardo Ferrer, Dairon, Dariel (La\_Masacre), Osiel, Juan Carlos y Dariel Corchado por contribuir en mi preparación como ingeniero y profesional.

A todos mis amigos y compañeros de universidad. En especial a Kelvis, Gabriel, a Noobster el calvo, Yanet, Lianne, Nayara, Noralis, Yaritza, el pina (Ichigo), Yannier, el Leo Martínez y Armando (Itachy).

A mis tutores Susana y Antonio por ayudarme a realizar este trabajo.

A todos mis familiares y profesores que de una forma u otra han ayudado a convertirme en la persona que soy.

A la UCI, mi segunda casa, la que nunca olvidaré.

## **Resumen**

Las nuevas tecnologías y herramientas para el desarrollo de aplicaciones web facilitan la ejecución de procesos de planificación y control. La automatización de estas actividades hace más viable el trabajo del ser humano actual, al agilizar las técnicas de gestión de la información. La Universidad de las Ciencias Informáticas (UCI) realiza diversas tareas que han sido informatizadas con el objetivo de alcanzar una mayor calidad en su ejecución y reducir la ocurrencia de errores. El Servicio de Guardia Obrera-Estudiantil es uno de los procesos que actualmente se planifica y gestiona de forma manual, mediante el empleo de la herramienta Microsoft Excel.

Debido a lo anteriormente expuesto, la presente investigación tiene como objetivo el desarrollo de una aplicación que garantice una correcta planificación y control del Servicio de Guardia Obrera-Estudiantil en la Facultad 4 de la UCI. Para el desarrollo de la investigación ha sido necesario el estudio de las principales características de las soluciones similares actualmente en uso. Se describen la arquitectura y las herramientas utilizadas para la implementación de la propuesta de solución. Se detallan la metodología utilizada y los artefactos generados en el proceso de desarrollo, así como las tecnologías empleadas. Finalmente se obtuvo una solución informática que responde a los objetivos planteados en la investigación.

**Palabras claves:** control, planificación, turno, zona, servicio de guardia

## Índice

<b>INTRODUCCIÓN</b> .....	<b>4</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>8</b>
1.1 CONCEPTOS GENERALES RELACIONADOS CON LA INVESTIGACIÓN .....	8
1.2 GESTIÓN DE PROCESOS UNIVERSITARIOS EN LA UCI .....	8
1.3 SOLUCIONES SIMILARES.....	9
1.3.1 <i>Turnex</i> .....	9
1.3.2 <i>Gestor web para el control de la Guardia Obrera de la Universidad</i> .....	10
1.3.3 <i>Plataforma GeoQ-Guardian</i> .....	11
1.3.4 <i>Sistema para la Gestión de la Guardia Obrera-Estudiantil en la Facultad 3</i> .....	12
1.3.5 <i>Valoración de las soluciones similares analizadas</i> .....	13
1.4 METODOLOGÍAS DE DESARROLLO DE SOFTWARE .....	13
1.4.1 <i>Scrum</i> .....	14
1.4.2 <i>XP</i> .....	14
1.4.3 <i>Valoración de las metodologías analizadas</i> .....	16
1.5 TECNOLOGÍAS, HERRAMIENTAS Y LENGUAJES DE DESARROLLO .....	16
1.5.1 <i>Framework de desarrollo</i> .....	16
1.5.1.1 <i>Zend Framework (ZF)</i> .....	17
1.5.1.2 <i>Symfony</i> .....	17
1.5.1.3 <i>Valoración de los framework analizados</i> .....	18
1.5.2 <i>Bootstrap</i> .....	18
1.5.3 <i>Lenguajes de programación</i> .....	18
1.5.3.1 <i>PHP</i> .....	19
1.5.3.2 <i>HTML</i> .....	20
1.5.3.3 <i>CSS</i> .....	21
1.5.3.4 <i>JavaScript</i> .....	21

1.5.4 Entorno de desarrollo integrado NetBeans.....	22
1.5.5 Sistema gestor de base de datos.....	22
1.5.5.1 PostgreSQL.....	23
1.5.5.2 MySQL.....	24
1.5.5.3 Valoración de los SGBD analizados.....	24
1.5.6 Servidor web Apache.....	24
1.5.7 Herramientas CASE de modelado con UML.....	25
1.5.7.1 Visual Paradigm para UML.....	25
1.5.7.2 UML.....	26
1.6 CONCLUSIONES DEL CAPÍTULO.....	26
<b>CAPÍTULO 2. EXPLORACIÓN Y PLANIFICACIÓN.....</b>	<b>28</b>
2.1 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	28
2.2 REQUISITOS NO FUNCIONALES DEL SISTEMA.....	29
2.3 USUARIOS RELACIONADOS CON EL SISTEMA.....	31
2.4 DIAGRAMA CONCEPTUAL DEL NEGOCIO.....	32
2.5 EXPLORACIÓN.....	32
2.5.1 Historias de usuario.....	33
2.6 PLANIFICACIÓN.....	35
2.6.1 Estimación de esfuerzo por HU.....	35
2.6.2 Plan de entregas.....	36
2.6.3 Iteraciones.....	37
2.7 DISEÑO.....	38
2.7.1 Tarjetas CRC.....	39
2.7.2 Patrones de diseño.....	40
2.7.2.1 GRASP.....	40
2.7.2.2 GoF.....	42
2.8 CONCLUSIONES DEL CAPÍTULO.....	42

<b>CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA.....</b>	<b>44</b>
3.1 ARQUITECTURA .....	44
3.2 ESTÁNDAR DE CODIFICACIÓN .....	45
3.3 BUNDLES DEL SISTEMA .....	46
3.4 TAREAS DE INGENIERÍA.....	47
3.5 FUNCIONAMIENTO DEL SISTEMA.....	48
3.6 PRUEBAS .....	49
3.6.1 Pruebas unitarias.....	49
3.6.2 Pruebas de aceptación .....	51
3.6.3 Resultados de las pruebas.....	52
3.7 CONCLUSIONES DEL CAPÍTULO .....	53
<b>CONCLUSIONES GENERALES .....</b>	<b>54</b>
<b>RECOMENDACIONES.....</b>	<b>55</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>55</b>
<b>ANEXOS .....</b>	<b>60</b>
ANEXO 1: HISTORIAS DE USUARIO.....	60
ANEXO 2: TARJETAS CRC .....	64
ANEXO 3: TAREAS DE INGENIERÍA .....	65
ANEXO 4: PRUEBAS DE ACEPTACIÓN .....	72

## Introducción

En las últimas décadas se ha hecho prácticamente imposible lograr una sociedad más eficaz, eficiente y competitiva sin hacer uso de las Tecnologías de la Información y las Comunicaciones (TIC), lo que ha provocado continuas transformaciones con el objetivo de aumentar la calidad de vida de los ciudadanos. Los novedosos avances de la informática como ciencia inciden de gran forma en la vida de los hombres debido a la informatización de muchos procesos que facilitan las labores cotidianas. Muchas han sido las herramientas creadas con la finalidad de viabilizar y optimizar numerosas actividades y como consecuencia se han vuelto prácticamente imprescindibles cuando se trata de alcanzar los resultados esperados con la mayor eficiencia y calidad posible. En Cuba se ha identificado la necesidad de introducir en la práctica social el uso de estas nuevas tecnologías. Bajo la guía del Programa Rector para la Informatización de la Sociedad Cubana se han priorizado esferas de gran importancia, tales como la educación, el deporte y la salud.

El Ministerio de Educación Superior cubano se encuentra inmerso en un grupo de transformaciones destinadas a mejorar el control, planificación, organización y ejecución de todos los procesos que son realizados actualmente de forma manual.

El Servicio de Guardia Obrera-Estudiantil (SGOE) es uno de los procesos que se realizan en la Universidad de las Ciencias Informáticas (UCI). El mismo está destinado a garantizar la seguridad de todos los medios materiales, estatales y personales, así como la integridad física de todos los que habitan en las distintas áreas de la institución. Este proceso es de vital importancia debido al impacto social, defensivo y organizativo que tiene para profesores, estudiantes y trabajadores.

Todas las facultades que integran la UCI realizan el proceso de guardia obrera-estudiantil. En la Facultad 4 este servicio se planifica de forma manual, con la utilización de la herramienta Microsoft Excel. Ello provoca que sea un proceso lento, tedioso y propenso a errores de planificación como: repetición de turnos y zonas, no poseer un buen control sobre la disponibilidad de los recursos humanos, no tener en cuenta si es residente o no en la universidad y ubicar mujeres en el tercer turno del SGOE.

En la UCI la Dirección de Seguridad y Protección es la encargada de velar por la correcta ejecución del servicio de guardia. La Facultad 4 tiene como obligación el envío de partes diarios y mensuales sobre el cumplimiento de esta actividad. Debido a esto se genera una gran cantidad de información no digitalizada,

gastos de recursos materiales y un significativo uso de la cantidad de tiempo empleada por el personal encargado de realizar la labor.

Para lograr hacer efectivo el proceso de planificación y control del SGOE, es imprescindible difundir la planificación con tiempo suficiente, preferentemente mayor a un mes. Para realizar esta tarea se debe tener en cuenta la existencia de factores como: la posibilidad de intercambios de turnos entre parejas de usuarios según su categoría, la necesidad de realizar reajustes a la agenda personal o plan de trabajo del potencial de guardia y el envío de la planificación vía correo electrónico.

La Facultad 4 cuenta con los medios informáticos, la infraestructura y el personal calificado para desarrollar una aplicación que permita la gestión del SGOE. Sin embargo, dicho proceso se realiza actualmente de forma manual, enviándose por correo electrónico la hoja de cálculo con la planificación realizada para el servicio de guardia.

A partir de la situación planteada surge como **problema de investigación**: ¿Cómo garantizar una correcta planificación y control del Servicio de Guardia Obrera-Estudiantil en la Facultad 4 de la Universidad de las Ciencias Informáticas?

Se precisa como **objeto de estudio** la gestión de procesos de planificación y control de la Universidad de las Ciencias Informáticas.

Se define para la investigación como **campo de acción** el proceso de planificación y control de la guardia obrera-estudiantil en la Facultad 4.

Para dar solución al problema antes expuesto se formula el siguiente **objetivo general**: desarrollar un sistema informático para garantizar una correcta planificación y control del Servicio de Guardia Obrera-Estudiantil en la Facultad 4 de la Universidad de las Ciencias Informáticas.

Para dar cumplimiento al objetivo general, se definen como **objetivos específicos**:

- Elaborar los referentes teóricos-metodológicos fundamentales relacionados con la temática a investigar para consultar y extraer información trascendental sobre el problema a investigar.
- Analizar las tecnologías, herramientas y sistemas existentes que gestionen la Guardia Obrera-Estudiantil, para estudiar la posibilidad de adaptarlos a los procesos de planificación y control de la Facultad 4.

- Diseñar el proceso a desarrollar en la Facultad 4; para emplear los SGOE seleccionados que contribuyan a una mayor calidad de los procesos de planificación y control de la facultad.
- Implementar el sistema de planificación y control del SGOE.
- Realizar pruebas al sistema implementado, para garantizar su correcto funcionamiento.

Se propone como **idea a defender** que el desarrollo de un sistema de gestión de la Guardia Obrera-Estudiantil en la Facultad 4, teniendo en cuenta las tecnologías y herramientas existentes, permitirá mayor control de los procesos de planificación de la facultad.

Las **tareas de la investigación** a realizar para dar solución a los objetivos específicos son:

- Revisión bibliográfica para conformar el estado del arte de la investigación mediante la búsqueda de información sobre soluciones similares aplicadas en diversos entornos.
- Diagnóstico del proceso actual de planificación y control del SGOE en la Facultad 4 de la UCI.
- Definición de los requerimientos que debe cumplir la propuesta de solución a desarrollar.
- Selección de la metodología, herramientas y tecnologías para guiar el proceso de desarrollo e implementar el sistema informático.
- Confección de los artefactos que propone la metodología de desarrollo de software seleccionada e implementación de la propuesta de solución.
- Realización de las pruebas de software a la propuesta de solución.

En la investigación se hizo uso de los métodos científicos determinados por el objetivo general y las tareas de investigación descritas anteriormente. A nivel teórico se utilizaron los métodos: **analítico-sintético**, para hacer un estudio bibliográfico amplio de la teoría existente alrededor del objeto de estudio, determinar las características que tendrá la propuesta de solución, y definir las tecnologías y herramientas más adecuadas para el desarrollo de la propuesta de solución; **histórico-lógico**, para tener en cuenta los aspectos fundamentales de la evolución histórica del proceso de planificación y control del servicio de guardia en la Facultad 4; **modelación**, el cual posibilita el esbozo de los diferentes diagramas y modelos del proceso de análisis y diseño para el desarrollo de la aplicación así como la validación de los campos obligatorios.

Para alcanzar cierto grado de organización y facilitar su estudio, el documento cuenta con una estructura capitular. Los capítulos que lo integran son los siguientes:

**Capítulo 1. Fundamentación teórica:** Se analizan los conceptos fundamentales acerca de la gestión y planificación de la guardia obrera-estudiantil (turno, zona, brigada y otros) y simultáneamente se tratan aspectos elementales relacionados con las tecnologías a emplear en el desarrollo de la aplicación. Se realiza un estudio detallado de la metodología que guiará el proceso de desarrollo de software.

**Capítulo 2. Exploración y planificación:** Se identifican los conceptos asociados al dominio del problema y los requisitos funcionales y las funcionalidades de la solución.

**Capítulo 3. Implementación y prueba:** Se describen los componentes asociados a la aplicación, se definen los estándares de codificación utilizados durante el desarrollo y se diseña una serie de casos de pruebas con el propósito de validar la aplicación, la evaluación de su ejecución y se describen los resultados obtenidos.

## Capítulo 1: Fundamentación Teórica

En el presente capítulo se realizará un estudio de la situación en la que se encuentra el Servicio de Guardia Obrera-Estudiantil, así como las definiciones de los conceptos relacionados con el mismo. Además, se describen las tecnologías usadas actualmente para el análisis, diseño e implementación del sistema sobre las cuales se apoya la propuesta de solución a desarrollar.

### 1.1 Conceptos generales relacionados con la investigación

Para lograr una correcta comprensión de la investigación realizada sobre el proceso de Guardia Obrero-Estudiantil es necesario definir conceptos claves que serán usados en la misma.

Según el Diccionario de la real Academia Española (DRAE) planificar es: *“trazar los planos para la ejecución de una obra, hacer plan o proyecto de una acción”*. (1). En la presente investigación se define turno como: *“ Orden según el cual se suceden varias personas en el desempeño de cualquier actividad o función”* (2). En relación a una zona se tiene como: *“ Parte de terreno o de superficie encuadrada entre ciertos límites, extensión considerable de terreno cuyos límites están determinados por razones administrativas, políticas, etc.”* (3). De acuerdo con el DRAE brigada es: *“ Conjunto de personas reunidas para dedicarlas a ciertos trabajos”* (4), por otra parte define asistencia como: *“ Acción de estar o hallarse presente”* (5). Por último, el DRAE define reporte como: *“Noticia, informe”* (6), al mismo tiempo, en el ámbito de la informática los reportes son informes que organizan y exhiben la información contenida en una base de datos. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios.

### 1.2 Gestión de procesos universitarios en la UCI

En la UCI se realizan tres procesos de gran importancia para la comunidad universitaria, los cuales son:

- Docencia.
- Extensión universitaria.
- Investigación y postgrados.

Con el objetivo de fortalecer la labor educativa, cultural y política-ideológica el proceso de extensión universitaria cuenta con un variado número de actividades. Se realizan eventos de gran importancia entre los que se destacan: Los Juegos Mella, Mi Beka más Bonita, Festival de Artistas Aficionados, SGOE,

entre otras. El SGOE está distribuido por las 7 facultades que integran la universidad, las cuales cuentan con zonas de vigilancia específicas para cada una.

La Facultad 4 posee dos zonas a proteger: Docente y Residencia. La zona Docente tiene dos comportamientos distintos uno para semana hábil (de lunes a viernes) y otro para el fin de semana (sábado y domingo). De lunes a viernes se cuenta con dos turnos del SGOE, los cuales cubren el horario de 10:00 pm a 7:00 am del día siguiente. El fin de semana se cuenta con cuatro turnos del SGOE para cubrir el horario de 8:00 am a 7:00 am del día sucesivo. Por otra parte, la zona Residencia cuenta con dos turnos del SGOE, los cuales cubrirán el horario de 11:00 pm a 7:00 am del día siguiente durante toda la semana (de domingo a sábado).

Para realizar la asignación de los turnos a ocupar durante la prestación de SGOE a cada persona, el responsable de la planificación debe tener en cuenta aspectos comunes para profesores y estudiantes. Se deben tener presente los miembros del potencial de guardia que se encuentran de vacaciones, los que no están disponibles debido a enfermedad u otros factores; la categoría que posee (estudiante o profesor); si es residente en la universidad (en caso de no serlo se le asigna un local para su hospedaje); si es mujer (se tendrá en cuenta si está embarazada o tiene hijos y que no se le asigne el último turno del SGOE).

### **1.3 Soluciones similares**

Como parte de la actual investigación se realizó un estudio de las soluciones existentes que resuelven un problema similar al descrito en el marco teórico-metodológico. Fueron analizados los siguientes sistemas:

#### **1.3.1 Turnex**

Es un software desarrollado por SOLEX para la gestión operativa detallada de turnos de trabajo de personas. Con su implementación es posible programar los turnos y/o horarios con anterioridad, lo que permite controlar las asistencias, días de descanso, estados y horas extras.

*“La herramienta permite integrar fácilmente los sistemas de nómina para la gestión del pago de remuneraciones, así como también, llevar bitácoras de trabajo y hojas de vida, representando una*

*excelente opción, principalmente para las empresas de servicios outsourcing<sup>1</sup> en recursos humanos o empresas industriales con personal interno en turnos rotativos.” (7)*

Características:

- Permite la reprogramación de turnos de los empleados.
- Posibilita un mejor aprovechamiento de los recursos humanos.
- Permite visualizar la información de la planificación de los turnos de guardia de los empleados.
- Permite el acceso concurrente de usuarios, debido a que es un producto 100% web.
- Permite la gestión de zonas.
- Permite controlar la asistencia de los guardias de seguridad.

TURNEX responde en gran medida a las necesidades actuales de la Facultad 4 en cuanto a la planificación y control de la Guardia Obrero-Estudiantil, sin embargo es un software privativo lo que implicaría en caso de que se quiera aplicar, un contrato con la empresa propietaria, conduciendo a un gasto monetario considerable.

### **1.3.2 Gestor web para el control de la Guardia Obrera de la Universidad**

Se desarrolla con el objetivo de mejorar el control y la obtención de reportes referente al proceso de Control de la Guardia Obrera en la UCI. Es independiente del sistema operativo donde se ejecute y presenta un requerimiento de hardware mínimo, brinda una amplia información sobre todo el tema perteneciente al control de la Guardia Obrera, minimiza el trabajo manual, agiliza el flujo de información y ahorra recursos a la universidad. El sistema cuenta con una interfaz web amigable y de fácil uso, lo que permite a los usuarios el disfrute de sus servicios.

Funcionalidades del sistema:

- Gestión de las postas.
- Gestión de los turnos de guardia.

---

<sup>1</sup> Outsourcing: uso de una firma externa especializada para proveer servicios en áreas que no forman parte de los procesos claves de la organización.

- Gestión de las áreas.
- Gestión de los trabajadores.
- Planificar la guardia de los trabajadores.
- Replanificar la guardia.
- Controlar la guardia planificada.
- Generar reportes e imprimir la información gestionada.

A pesar de ser una aplicación desarrollada en la UCI no se cuenta con el código fuente de la misma. Fue implementada sin el uso de un framework de desarrollo lo que deja una brecha abierta a la seguridad del sistema y al uso de malas prácticas en su construcción como son: poca reusabilidad del código y que los programadores desarrollen de forma distinta, posibilitando que se implemente lo que alguien ya hizo. Para realizar la planificación de la guardia no tiene en cuenta la categoría estudiante, solo los trabajadores.

### **1.3.3 Plataforma GeoQ-Guardian**

La plataforma GeoQ-Guardian fue desarrollada por el centro de Geo-Informática y Señales Digitales (GEYSED) perteneciente a la UCI. Esta plataforma surge por la necesidad de agilizar y optimizar el proceso de planificación y control de la guardia en la universidad.

Funcionalidades del sistema:

- Gestión de las postas.
- Gestión de los turnos de guardia.
- Gestión de las zonas.
- Gestión del personal que realiza la guardia.
- Planificar la guardia.
- Modificar la planificación de la guardia.
- Controlar la guardia.
- Generar reportes e imprimir la información gestionada.

GeoQ-Guardian es una aplicación de escritorio y bastante abarcadora en relación a las necesidades de la Facultad 4, sin embargo no es apropiada para la gestión de la Guardia Obrero-Estudiantil. Esto se debe a que para planificar el servicio de guardia en cada área de la universidad la accesibilidad se ve afectada, debido a que solo pueden interactuar con la aplicación el personal del área donde está instalada. Además, el sistema cuenta con un tiempo de respuesta que provoca una pérdida de eficiencia debido al cúmulo de información. Por otra parte el proceso de mantenimiento es complejo, debido a que se requiere de actualización manual en cada una de las estaciones de trabajo.

Debido a que la forma de diseño está centrada en un único usuario local, la capacidad de autenticación y autorización no es posible. Además, este software no es multiplataforma, solo es funcional en Windows, lo que es una característica importante en un entorno donde el software libre cuenta con una gran aceptación.

#### **1.3.4 Sistema para la Gestión de la Guardia Obrera-Estudiantil en la Facultad 3**

Este sistema es desarrollado en la Facultad 3 de la UCI para gestionar la planificación del SGOE en el curso 2012-2013. Además, esta aplicación pretende facilitar el monitoreo y control del proceso de gestión de la Guardia Obrera-Estudiantil. Es un sistema multiplataforma.

Principales funcionalidades:

- Gestionar postas.
- Gestionar turnos.
- Planificar la guardia.
- Eliminar planificación de la guardia.
- Imprimir planificación.
- Gestionar cambios de guardia.

Es válido aclarar que el proceso del SGOE, con respecto al momento en que fue desarrollada esta solución, ha sufrido cambios como: modificación del horarios de los turnos de guardia, reducción de la cantidad de postas a cubrir por la Facultad 4 (se disminuyó de diez postas, de la manzana 30 a la 10 A, a

una posta en las manzanas 30 y 31) y un aumento en el horario de los turnos en el docente (media hora más), conllevando a la necesidad de disminuir la cantidad de profesores en la residencia.

### **1.3.5 Valoración de las soluciones similares analizadas**

Al realizar el análisis de estos sistemas que brindan soluciones similares para la planificación y control del SGOE se puede apreciar que no se ajustan totalmente a las necesidades de la Facultad 4, sin embargo con su estudio se obtuvieron nuevas ideas y funcionalidades que optimizarán el proceso de planificación y control del SGOE. Estas funcionalidades son:

- Planificar la guardia.
- Controlar asistencia.
- Generar reportes.

### **1.4 Metodologías de desarrollo de software**

El conjunto de procedimientos, técnicas y ayudas a la documentación para desarrollar una aplicación informática es denominado metodologías de desarrollo de software. Estas brindan una guía paso a paso de todas las actividades propuestas para lograr el producto deseado. Además, indican las personas que deben participar en el desarrollo de las actividades y el papel que desempeñan cada una. También, detallan la información necesaria para comenzar una actividad y la que se debe producir como resultado de la misma.

*“Las metodologías tradicionales propician las buenas prácticas que existen dentro de la Ingeniería de Software; sin embargo, requieren de mucha disciplina para seguir con el riguroso proceso que éstas conllevan. Contribuyen positivamente al ser incrementales e iterativas; promueven la asignación de roles dentro del equipo de desarrollo, facilitan la división del sistema en varios subsistemas y fomentan la reutilización de componentes. De manera general, las metodologías tradicionales consideran la importancia de documentar el sistema, permitiendo así, entender, extender y mantener el software.” (8)*

*“Las metodologías ágiles presentan respuestas rápidas al cambio y son flexibles, generan poca documentación y no hacen uso de métodos formales. Buscan la satisfacción del cliente y la entrega temprana del software incremental; equipos de desarrollo pequeños, con alta motivación y un mínimo de*

*productos de trabajo. Las directrices de desarrollo resaltan la entrega sobre el análisis y diseño, la comunicación activa y constante entre los desarrolladores y el cliente.” (8)*

A partir de lo analizado anteriormente se selecciona el enfoque ágil para guiar el proceso de desarrollo. La elección estuvo determinada por el poco tiempo de desarrollo con que se cuenta, cantidad de miembros del equipo de trabajo y la existencia de un alto riesgo de realizar cambios en los requisitos. Dentro de las metodologías ágiles se encuentran Crystal Clear, Agile Unified Process (AUP), Lean Software Development (LSD), Programación Extrema (por su nombre en inglés XP), Scrum, Kanban, entre otras. Debido a que resultan más adaptables para proyectos a corto plazo y son de las más usadas, se seleccionó Scrum y XP, como metodologías a analizar para poder determinar cuál guiará el proceso de desarrollo de la propuesta de solución.

#### **1.4.1 Scrum**

*“Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.” (9)*

Scrum indica la realización de entregas parciales y regulares del producto final, basándose en la prioridad del beneficio que aporten al cliente. Esta metodología está especialmente enfocada para proyectos en entornos complejos donde es necesario obtener resultados pronto y los requisitos son cambiantes o poco definidos. Además, es la más adecuada en ambientes donde la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se entrega al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan; la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, la moral de los equipos es baja y la rotación alta; cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar con la utilización de un proceso especializado en el desarrollo de producto.

#### **1.4.2 XP**

Como clave para el éxito del desarrollo de software la metodología ágil XP plantea potenciar las relaciones interpersonales, promover el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y

por propiciar un buen clima de trabajo. La realimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes y la simplicidad en las soluciones implementadas son las bases de esta metodología. Debido a esto, XP se puede definir como adecuada para proyectos con requisitos imprecisos y muy cambiantes.

Necesidades del entorno XP:

- Semana de 40 horas de trabajo.
- Presencia de los desarrolladores en un mismo local: consigue que la mayoría de la documentación se pueda suprimir sin mayores consecuencias porque el proceso de comunicación se realiza directamente, de forma hablada.
- Todos los días comienzan con una reunión de apertura: agiliza la comunicación entre los integrantes del proyecto y estos están constantemente informados sobre el estado del trabajo.
- Alta velocidad de trabajo: debido a la realización de proyectos a corto plazo.

Características fundamentales de XP (10):

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas: se basan en las pruebas realizadas a los principales procesos con el objetivo de detectar futuros errores.
- Integración del equipo de programación con el cliente o usuario: se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores: antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código: reescribir ciertas partes del código para aumentar su legibilidad y mantenimiento pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Uso de Metáforas: la comunicación fluida es uno de los valores más importantes de XP, el hecho de incorporar al equipo una persona que represente los intereses del negocio y otras prácticas son valiosas porque potencian enormemente la comunicación. Para conseguir que la comunicación sea fluida es imprescindible, entre otras cosas, utilizar el vocabulario del negocio.

### 1.4.3 Valoración de las metodologías analizadas

Con la información anteriormente planteada se concluyó que es más factible hacer uso de la metodología XP debido a las ventajas que brinda para equipos de trabajo con poco personal y con una buena relación con el cliente. También, esta metodología permite: empleo de estándares de codificación y pruebas continuas, los cambios se implementan rápidamente y el diseño del software es sencillo y libre de complejidad.

## 1.5 Tecnologías, herramientas y lenguajes de desarrollo

Los continuos avances tecnológicos demandan constantemente aplicaciones más rápidas, ligeras y robustas que permitan ser usadas sin importar el lugar u horario. Como resultado de las ventajas que ofrece la red, se comenzaron a desarrollar aplicaciones de servicio y de solución de problemas entre las que se encuentran las aplicaciones Web.

*“Las aplicaciones web permiten la generación automática de contenido, la creación de páginas personalizadas según el perfil del usuario o el desarrollo del comercio electrónico. Además, una aplicación web permite interactuar con los sistemas informáticos de gestión de una empresa, como puede ser gestión de clientes, contabilidad o inventario, a través de una página web. Las aplicaciones web se encuadran dentro de las arquitecturas cliente/servidor: un ordenador solicita servicios (el cliente) y otro está a la espera de recibir solicitudes y las responde (el servidor).” (11)*

### 1.5.1 Framework de desarrollo

*“Un framework es una abstracción en la que cierto código común provee una funcionalidad genérica que puede ser sobrescrita o especializada de forma selectiva por medio de código con funcionalidad específica. Un framework es una solución incompleta (no funcional) pero concreta (a diferencia de los estilos arquitectónicos o los patrones de diseño) a un problema recurrente bien conocido.” (12)*

Los frameworks o marcos de trabajo, proveen plantillas que definen el funcionamiento de las aplicaciones, al permitir el manejo y control de las mismas sin tener que escribir mucho código. Las plantillas ofrecidas por los marcos de trabajo se pueden adaptar fácilmente a las diferentes necesidades del desarrollador. Además, los artefactos o módulos que estos brindan sirven de base para la organización y desarrollo del software.

### 1.5.1.1 Zend Framework (ZF)

Es un framework para el desarrollo de aplicaciones web y servicios web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Trabaja con PHP 5, es de código abierto, está orientado a objetos y cada uno de sus componentes está construido con una baja dependencia de otros componentes.

*“Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de ZF conforman un potente y extensible framework de aplicaciones web al combinarse. ZF ofrece una robusta implementación Modelo Vista-Controlador y una abstracción de base de datos fácil de usar.”* (13)

### 1.5.1.2 Symfony

*“Es un proyecto PHP de software libre que permite crear aplicaciones y sitios web rápidos y seguros de forma profesional. Es de tipo full-stack<sup>2</sup> construido con varios componentes independientes creados por el proyecto Symfony.”* (14)

Principales características:

- Desarrollo muy activo, puesto que incluye constantemente mejoras que combinan la flexibilidad con la rapidez de ejecución.
- Amplia documentación.
- Reutilización muy activa de usuarios: la comunidad de usuarios crece cada día y ofrece un gran soporte de forma gratuita.
- Flexibilidad tanto en el diseño global del framework, como en su sistema de configuración y en los plugins<sup>3</sup>.

El Mapeador de Objeto Relacional (ORM) Doctrine 2 es usado por defecto en Symfony 2. Doctrine tiene como principal característica el bajo nivel de configuración que requiere para comenzar un proyecto. El programador puede especificar relaciones y adicionar funcionalidades comunes para las clases generadas

---

<sup>2</sup> Full-stack: permite crear aplicaciones complejas desde cero.

<sup>3</sup> Plugins: aplicaciones que se relacionan con otras para aportarles una función nueva y generalmente muy específica.

a partir de una base de datos creada. Además, posee la habilidad de escribir consultas a la base de datos a partir de la programación orientada a objetos, llamada Doctrine Query Language (DQL).

### **1.5.1.3 Valoración de los framework analizados**

Debido al conocimiento que se posee sobre el framework Symfony y sus ventajas para el desarrollo de aplicaciones web se seleccionó para el desarrollo de la propuesta de solución. La versión a utilizar será 2.5, que destaca sobre las versiones anteriores por garantizar una mayor seguridad, contar con formularios integrados con Doctrine2, brindar una amplia flexibilidad al permitir realizar cambios a todo lo creado con el framework y proporcionar un mayor rendimiento en su ejecución.

### **1.5.2 Bootstrap**

Es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo.

*“Aun ofreciendo todas las posibilidades que ofrece Bootstrap a la hora de crear interfaces web, los diseños creados con Bootstrap son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El Framework trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos.” (15)*

Debido a la existencia de conocimientos previos y su facilidad de uso, fue la tecnología escogida para el desarrollo de la propuesta de solución. Se utilizara la versión 3.3.2.

### **1.5.3 Lenguajes de programación**

Los lenguajes de programación están diseñados para crear programas a través de algoritmos que son procesados por computadoras. Brindan la oportunidad de realizar instrucciones a una aplicación informática. Construir proyectos con el empleo de un lenguaje de programación conlleva al uso de símbolos, reglas sintácticas y semánticas que definen su estructura y significado.

### 1.5.3.1 PHP

*“Es un lenguaje de scripting que permite la generación dinámica de contenidos en un servidor web. El significado de sus siglas es HyperText Preprocessor. Entre sus principales características cabe destacar su potencia, su alto rendimiento, su facilidad de aprendizaje y su escasez de consumo de recursos.” (16)*

Dentro de una página web se puede incrustar código PHP que se ejecutará siempre que esta sea visitada. El código PHP es interpretado en el servidor web y genera código HTML y otros contenidos que el visitante puede ver.

Características:

- Alto rendimiento: PHP es muy eficiente, mediante el uso de un único servidor, puede soportar millones de accesos al día.
- Interfaces para una gran cantidad de sistemas de base de datos diferentes: dispone de una conexión propia a todos los sistemas de base de datos. Además de MySQL, puede conectarse directamente a las bases de datos de PostgreSQL, mySQL, Oracle, dbm, filepro, Hyperwave, Informix, InterBase y Sybase, entre otras. El uso de ODBC (Estándar de conectividad abierta de base de datos) permite establecer una conexión a cualquier base de datos que suministre un controlador ODBC.
- Bibliotecas incorporadas para muchas tareas web habituales: Como se ha diseñado para su uso en la web, PHP incorpora una gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la web. Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo con unas pocas líneas de código.
- Bajo coste: PHP es gratuito.
- Facilidad de aprendizaje y uso: La sintaxis de PHP se basa en otros lenguajes de programación, principalmente en C y Perl.

Se seleccionó el lenguaje PHP debido a que el framework a utilizar es Symfony y está escrito en este lenguaje. La versión a utilizar será 5.5.4.

### 1.5.3.2 HTML

*“Hyper Text Markup Language es un lenguaje que sirve para escribir hipertexto, es decir, documentos de texto presentado de forma estructurada, con enlaces (links) que conducen a otros documentos o a otras fuentes de información (por ejemplo bases de datos) que pueden estar en tu propia máquina o en máquinas remotas de la red. Todo ello se puede presentar acompañado de cuantos gráficos estáticos o animados y sonidos seamos capaces de imaginar.” (17)*

HTML es un lenguaje muy sencillo que nos permite preparar documentos Web en los que se insertan en el texto de los mismos una serie de marcas (tags) que controlan los diferentes aspectos de la presentación y comportamiento de sus elementos.

HTML5 es la última evolución de la norma que define HTML. El término representa dos conceptos diferentes:

- *“Se trata de una nueva versión del lenguaje HTML, con nuevos elementos, atributos y comportamientos.” (18)*
- *“Conjunto más amplio de tecnologías que permite a los sitios Web y las aplicaciones más diversas y de gran alcance. Este conjunto se le llama HTML5 y amigos, a menudo reducido a sólo HTML5.” (18)*

HTML 5 propone estándares para cada aspecto de la web y también un propósito claro para cada una de las tecnologías involucradas.

El hecho de ser este lenguaje un estándar establecido por la World Wide Web Consortium (W3C) determinó su elección para realizar el maquetado de la propuesta de solución. La versión a utilizar será HTML 5. La elección de esta versión estuvo determinada por la utilización del framework de diseño Bootstrap 3.3.2, la cual hace uso de HTML 5. Además, en comparación a versiones anteriores el código es más sencillo y ligero, permitiendo que las páginas carguen más rápido, permite la validación de campos sin usar JavaScript a partir de una expresión regular y no es necesario el cierre de algunas etiquetas como: img, br, hr, input, entre otras.

### 1.5.3.3 CSS

*“Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.” (19)*

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo en infinidad de dispositivos diferentes.

*“Se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.” (19)*

La elección de CSS 3 estuvo determinada por la utilización del framework Bootstrap 3.3.2, el cual emplea esta versión. Además, CSS 3 incluye los bordes redondeados, textos con sombra, mejor manejo de tablas y de la interfaz gráfica. Por último se tuvo en cuenta que el mismo es un estándar determinado por W3C, por lo que se usará para escribir el estilo visual de la propuesta de solución.

### 1.5.3.4 JavaScript

*“JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.” (20)*

El estándar DOM (Document Object Model) fue creado por la W3C para evitar incompatibilidades a la hora de integrar el código JavaScript dentro de las páginas web.

Debido al hecho de utilizar la librería JQuery, la cual está escrita en este lenguaje, se seleccionó JavaScript como lenguaje de script del lado del cliente.

#### **1.5.4 Entorno de desarrollo integrado NetBeans**

Es un IDE de código abierto que posee las herramientas necesarias para el desarrollo de aplicaciones para teléfonos móviles, web y de escritorio. Utiliza lenguajes de programación como Java, C, C++, además de lenguajes dinámicos como: PHP, JavaScript, Groovy y Ruby. Es multiplataforma, puede ser usado en Windows, Linux, MAC OS X y Solaris. Es de fácil uso e instalación.

Principales características:

- Creación de proyectos PHP: Provee una estructura para los proyectos que se pueden crear junto a este IDE, así como propone un esqueleto para la organización del código fuente.
- Integración con Symfony y Zend Framework: Posibilita dejar de lado la consola de comandos de Symfony y centrarse en desarrollar en el IDE.
- Integración con PHP Unit Testing: Permite crear test con PHPUnit, para diferentes funciones, realizar la comprobación y ver todos los resultados.
- Editor de código fuente: El editor de PHP es mucho más ágil y robusto. Contiene todo lo que provee la versión de PHP 5.3.
- Integración con MySQL: Posee una integración completa en términos de administración básica y avanzada de MySQL.
- Integración con Sistemas de Control de Versiones: Es una condición necesaria para los proyectos la integración con sistemas como SVN, Git y CVS.

El IDE de NetBeans permite inspeccionar el código HTML generado sin tener que cambiar a un navegador, debido a que ofrece la línea de comandos de depuración para PHP.

Debido a la experiencia en el trabajo con esta herramienta y que brinda un amplio soporte al desarrollo de aplicaciones web con el empleo de Symfony, se seleccionó para el desarrollo de la propuesta de solución. La versión a utilizar es la 8.0.

#### **1.5.5 Sistema gestor de base de datos**

Un Sistema Gestor de Base de Datos (SGBD) tiene como objetivo servir de interfaz entre las bases de datos, las aplicaciones y el usuario. Permite organizar, introducir y recuperar la información de las bases

de datos. De los distintos tipos de gestores de bases de datos que existen el modelo relacional es el más usado para PC's. Algunos ejemplos de SGBD son PostgreSQL y MySQL.

#### 1.5.5.1 PostgreSQL

Es un servidor de bases de datos relacional orientada a objetos de software libre que soporta casi toda la sintaxis SQL. Ofrece muchas características modernas tales como: consultas complejas, integridad referencial, integridad transaccional, control de concurrencia multi-versión, entre otras.

Características generales (21):

- Extensible: El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.
- Multiplataforma: PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable).
- Diseñado para ambientes de alto volumen: PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.

Ventajas:

- Ampliamente popular. Ideal para tecnologías web.
- Fácil de administrar.
- Su sintaxis SQL es estándar y fácil de aprender.
- Multiplataforma.
- Capacidades de replicación de datos.
- Soporte empresarial disponible.

### 1.5.5.2 MySQL

Creado por la empresa sueca MySQL AB, es un sistema para la administración de bases de datos relacionales. Permite manejar los datos almacenados en tablas de una forma eficiente y segura. El lenguaje de programación SQL es usado para gestionar una base de datos relacional. MySQL usa la licencia GPL (GNU General Public License) para aplicaciones no comerciales y su código fuente está accesible a cualquiera.

*“Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.” (22)*

### 1.5.5.3 Valoración de los SGBD analizados

El carácter libre y los conocimientos previos con los que cuenta el equipo de trabajo determinaron la elección de PostgreSQL para ser el gestor de bases de datos en el desarrollo de la propuesta de solución.

### 1.5.6 Servidor web Apache

Un servidor web en internet es un ordenador remoto que provee los datos solicitados por los navegadores de otras computadoras. En redes locales es el software que permite configurar una computadora como servidor para facilitar el acceso a la red y sus recursos. Un servidor almacena información en forma de páginas web y lo entregan en formato HTML a petición de los clientes a través del protocolo HTTP. Uno de los servidores web más usado en el mundo es Apache.

El servidor web Apache no solo funciona en la mayoría de las versiones de Unix, también en Windows 2000/NT/9x y en muchos otros sistemas operativos. Es de código fuente abierto.

Características:

- *“Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP.” (23)*
- *“Multiplataforma.” (23)*

- *“Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.” (23)*
- *“Extensible: El hecho de ser modular ha permitido el desarrollo de diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.” (23)*

Apache es el servidor web a utilizar debido a su amplia difusión y facilidad de configuración. La versión elegida será la 2.4.9.

### **1.5.7 Herramientas CASE de modelado con UML**

Las herramientas de ingeniería de software asistida por computadora (CASE), son aplicaciones computacionales en conjunto que soportan y ayudan al proceso de análisis y desarrollo de Software. Durante todo el Ciclo de Vida del producto sirven a los analistas de sistemas, ingenieros de software y desarrolladores.

#### **1.5.7.1 Visual Paradigm para UML**

Es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia al equipo de trabajo en la construcción del software.

Las ventajas que proporciona Visual Paradigm para UML son:

- Dibujo. Facilita el modelado de UML, debido a que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, debido a que la misma se ajusta al estándar soportado por la herramienta.
- Corrección sintáctica. Controla que el modelado con UML sea correcto.
- Coherencia entre diagramas. Al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas y evitar las duplicidades.
- Integración con otras aplicaciones. Permite integrarse con otras aplicaciones, como herramientas ofimáticas, aspecto que aumenta la productividad.
- Trabajo multiusuario. Permite el trabajo en grupo, al propiciar herramientas de compartición de trabajo.

- Reutilización. Facilita la reutilización, debido a que se dispone de una herramienta centralizada donde se encuentran los modelos utilizados para otros proyectos.
- Generación de código. Permite generar código de forma automática, lo que reduce los tiempos de desarrollo, así como evitar errores en la codificación del software.
- Generación de informes. Permite generar diversos informes a partir de la información introducida en la herramienta.

### 1.5.7.2 UML

*“UML está diseñado para su uso con software orientado a objetos, y tiene un uso limitado en otro tipo de cuestiones de programación. Se compone de muchos elementos de esquematización que representan las diferentes partes de un sistema de software. Los elementos UML se utilizan para crear diagramas, que representa alguna parte o punto de vista del sistema.” (24)*

UML posibilita el establecimiento de una serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código y sus beneficios son directamente proporcionales a la complejidad del sistema que se desea. Intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos se crea una documentación que cualquier desarrollador con conocimientos de UML será capaz de entender.

Debido a que soporta el ciclo de vida completo del desarrollo del software a obtener y que permite el modelado de sistemas con el empleo de técnicas orientadas a objeto se seleccionó como lenguaje de modelado el UML con la utilización de la herramienta Visual Paradigm. La versión a utilizar es la 8.0.

### 1.6 Conclusiones del capítulo

Con el análisis realizado sobre el proceso de informatización del SGOE y el estado del arte del mismo, se lograron definir los conceptos importantes asociados a la investigación. El estudio de las soluciones similares arrojó nuevas ideas para el desarrollo del sistema y ayudó a la comprensión de los aspectos característicos de este tipo de aplicaciones, además de aportar algunas funcionalidades que responden a necesidades de los clientes que serán tomadas en cuenta para la propuesta de solución.

El estudio de las herramientas y tecnologías a utilizar en la propuesta de solución permitieron seleccionar la metodología XP para guiar el proceso de desarrollo. Se definieron las tecnologías para usar del lado del

servidor. Como lenguaje se utilizara PHP 5.5.4, mediante el uso del framework de desarrollo Symfony 2.5. En el lado del cliente será utilizado como lenguaje de maquetado HTML 5. Los estilos visuales serán manejados con CSS 3 a través del framework Bootstrap. Como lenguaje de scripting se utilizará JavaScript a través de la librería JQuery 2.1.3. Para almacenar la información se decidió usar PostgreSQL 9.3 como gestor de bases de datos.

## Capítulo 2. Exploración y planificación

En el presente capítulo se documentarán las fases de exploración y documentación. Para dar respuesta al problema planteado se describirá la propuesta de solución. Además se especificarán los roles que interactuarán con el sistema y debido al tiempo con que cuenta el equipo de trabajo se generarán los artefactos necesarios para el desarrollo del sistema según la metodología XP, tales como: historias de usuarios (HU)<sup>4</sup>, estimación de esfuerzo por historias de usuarios, plan de entregas e iteraciones.

### 2.1 Descripción de la propuesta de solución

Debido a que las soluciones similares estudiadas en el capítulo anterior no satisfacen las necesidades de la Facultad 4 el equipo de desarrollo de la presente investigación brinda como propuesta de implementación un sistema cuya finalidad sea realizar una correcta planificación y control del SGOE en la Facultad 4. Dentro del sistema los usuarios desempeñarán los roles: Super-Administrador, Administrador y Estándar. El acceso inicial se realizará mediante el usuario con rol Super-Administrador, que será el encargado de entrar al sistema los datos de los usuarios.

El usuario con rol Super-Administrador podrá:

- Definir los roles de cada usuario ingresado al sistema.
- Ingresar a la base de datos del sistema los datos del personal a tener en cuenta en la planificación de la guardia.

El usuario con rol Administrador será el encargado de:

- Gestionar la planificación del SGOE.
- Controlar la asistencia.
- Crear las brigadas de estudiantes.
- Cambiar el estado de disponibilidad de cada usuario en el sistema.
- Exportar a formato PDF<sup>5</sup> la planificación general del SGOE y la asistencia diaria.

---

<sup>4</sup> HU: Describen detalladamente los requisitos funcionales del software.

<sup>5</sup> PDF: Portable Document Format (formato de documento portátil).

- Mostrar planificación del SGOE.

El usuario con rol Estándar podrá:

- Mostrar su planificación del SGOE.
- Mostrar su asistencia.
- Realizar intercambios de turnos en el SGOE.
- Autorizar intercambio de turnos en el SGOE.
- Mostrar información referente al SGOE.

Las acciones que realicen los usuarios con rol Estándar podrán ser realizadas por los usuarios con rol Administrador y Super-administrador debido a que estos roles fueron definidos como una especialización del rol Estándar. El despliegue del sistema se realizará en la PC servidor del sitio web Octavitos con la aprobación del Departamento de Seguridad Informática de la Facultad 4 y su acceso será mediante un enlace que se incluirá en el sitio antes mencionado. Además, en esta computadora se encontrarán al mismo tiempo, el servidor de bases de datos y de aplicaciones.

## **2.2 Requisitos no funcionales del sistema**

*“Los requerimientos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema.” (25)*

Los requisitos no funcionales son cualidades o propiedades que el producto debe tener y que pudieran limitar al sistema. Estas propiedades convierten al producto en confiable y rápido. Además, estos requisitos garantizan características esenciales en la aplicación como el tiempo de respuesta y la capacidad de almacenamiento.

### **Requerimientos de software:**

Debe poder visualizarse en los navegadores:

- Mozilla Firefox 5.0 o superior.

- Internet Explorer 9 o superior.
- Google Chrome 10.0 o superior.
- Opera 11.0 o superior.
- Safari 5.0 o superior.

**Requerimientos de hardware:**

**Para el cliente:**

- Tarjeta de red.
- Al menos 512 MB de RAM.
- Procesador Dual Core 1.3 GHz como mínimo.

**Para el servidor:**

- Tarjeta de red.
- Al menos 1GB de RAM.
- Procesador Core 2 Duo 2.4 GHz como mínimo.

**Requerimientos de Seguridad:**

- La única forma de acceder al sistema es después de autenticarse (usuario y contraseña).
- Los usuarios que pueden acceder son los que estén en el dominio UCI y registrados en la base de datos.

**Requerimientos de Usabilidad:**

- La interfaz de usuario de la aplicación debe ser sencilla, intuitiva, que le permita al usuario que interactúa con el sistema realizar cualquier labor en el menor tiempo posible y con pocas acciones.

**Requerimientos de Rendimiento:**

- Los tiempos de respuesta del sistema en el procesamiento de datos no deben exceder los 10 segundos.

**Requerimientos Políticos Culturales:**

- El sistema no debe contener palabras en otros idiomas.
- El sistema debe respetar los términos empleados normalmente por los especialistas en el tema de la esfera que se automatiza.

**Requerimientos Legales:**

- Las herramientas de desarrollo son libres.
- Reconocido y autorizado por instancias superiores tales como la Dirección General de Producción de la UCI (Facultad 4).
- Documentación legal de uso como Declaración de Autoría.

**Requerimientos de Confiabilidad:**

- Los datos que se obtendrán de la aplicación deben ser 100% precisos.

**2.3 Usuarios relacionados con el sistema**

Los usuarios del sistema son personas que se conectan para hacer uso de los servicios que este les proporciona. Dentro de ellos se pueden distinguir diferentes perfiles que definen los privilegios con los que se contará. Además, las restricciones lógicas definidas para cada uno de ellos garantizan la accesibilidad y una mayor conservación de las propiedades que posee el sistema.

**Tabla 1: Usuarios del sistema**

<b>Usuario</b>	<b>Descripción</b>
<b>Super-Administrador</b>	Ingresa en el sistema los datos de los usuarios y define el rol de cada usuario.
<b>Administrador</b>	Modifica y realiza todas las funcionalidades definidas para el sistema, al controlar toda la información de los usuarios.

<b>Estándar</b>	Consulta los datos referentes a la planificación de la guardia, registro de asistencia e intercambiar turnos de guardia.
-----------------	--

### 2.4 Diagrama conceptual del negocio

Un diagrama conceptual del negocio es un artefacto que está construido bajo las reglas de UML. Esto sucede durante la concepción del proyecto informático. Además, este modelo puede ser usado para capturar y expresar el dominio adquirido sobre el negocio.

Para un mejor entendimiento de la propuesta de solución a desarrollar se generó el diagrama conceptual luego de identificar los conceptos y objetos relacionados con la planificación y control del SGOE en la Facultad 4, definido de la siguiente manera:

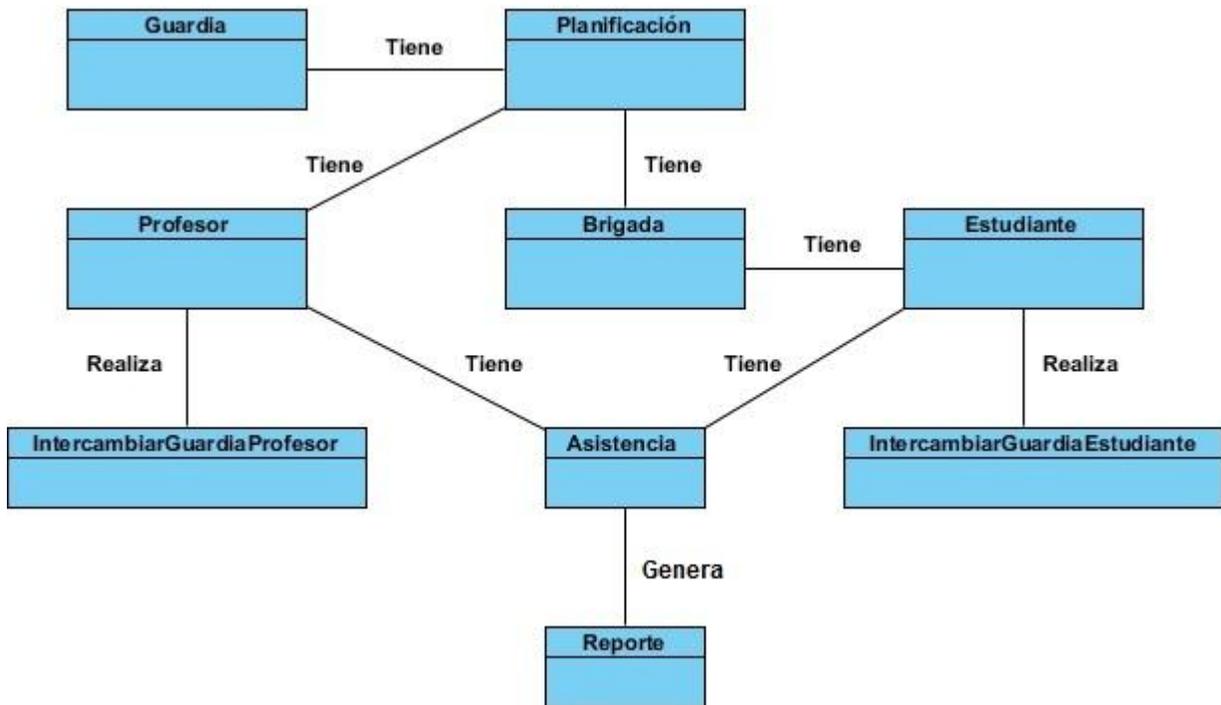


Ilustración 1: Diagrama conceptual

### 2.5 Exploración

Las HU se precisan en la fase de exploración de la metodología XP. Constituyen las descripciones de las funcionalidades con que debe contar el sistema y se describen en una plantilla las características que

deben ser agregadas al mismo. A la vez, el equipo de desarrollo se familiariza con las tecnologías y herramientas que serán usadas en el transcurso del desarrollo del proyecto.

La exploración: *“Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuarios”. Los programadores estiman los tiempos de desarrollo en base a esta información.”* (26)

### **2.5.1 Historias de usuario**

En XP las HU son la técnica para especificar los requisitos del software. Son descripciones cortas escritas en el lenguaje del usuario. Proporcionan una estimación del riesgo y del tiempo que conllevará su implementación. Son artefactos muy flexibles, prestos a sufrir cambios en cualquier momento o a admitir nuevos elementos, por lo que el nivel de detalle es el mínimo posible. Las HU son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración determinada.

Funcionalidades del sistema:

1. Autenticar usuario.
2. Gestionar datos del usuario.
3. Crear brigadas del SGOE.
4. Generar planificación del SGOE.
5. Generar PDF.
6. Mostrar planificación del SGOE.
7. Mostrar mi planificación del SGOE.
8. Mostrar reporte de cumplimiento del SGOE.
9. Controlar asistencia del SGOE.
10. Buscar integrantes en el SGOE para solicitar intercambio de turnos.
11. Intercambiar turnos en el SGOE.
12. Listar intercambio de turnos en el SGOE.

13. Gestionar rol.

14. Importar usuarios.

**Tabla 2: Plantilla HU**

Historia de usuario	
<b>Número:</b> Número de la HU.	<b>Usuario:</b> El usuario del sistema que utiliza o protagoniza la HU.
<b>Nombre:</b> Nombre de la HU.	
<b>Prioridad en el negocio:</b> Cuan importante es para el cliente o equipo de desarrollo. Alta, Media o Baja.	<b>Riesgo de desarrollo:</b> Nivel de dificultad para el desarrollador. Alto, Medio o Bajo.
<b>Puntos de estimación:</b> Tiempo estimado para realizar la actividad.	<b>Iteración:</b> Iteración a la que corresponde.
<b>Descripción:</b> Breve reseña de la HU, describiendo las acciones del usuario.	

HU para la realización de la aplicación, las restantes se encuentran en los anexos.

**Tabla 3: Historia de Usuario No. 1**

Historia de usuario	
<b>Número:</b> 1	<b>Usuario:</b> Estándar, Administrador y Súper-Administrador.
<b>Nombre:</b> Autenticar usuario.	
<b>Prioridad en el negocio:</b> Alta.	<b>Riesgo de desarrollo:</b> Medio.
<b>Puntos de estimación:</b> 0.5	<b>Iteración:</b> 1
<b>Descripción:</b> El sistema comprueba que los datos introducidos por los usuarios para acceder a este son correctos.	

**Tabla 4: Historia de Usuario No. 2**

Historia de usuario	
<b>Número:</b> 2	<b>Usuario:</b> Súper-Administrador.

<b>Nombre:</b> Gestionar datos del usuario.	
<b>Prioridad en el negocio:</b> Alta.	<b>Riesgo de desarrollo:</b> Alto.
<b>Puntos de estimación:</b> 1	<b>Iteración:</b> 1
<b>Descripción:</b> El sistema debe insertar, modificar y mostrar todos los usuarios en el sistema.	

## 2.6 Planificación

*“En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses.” (27)*

La planificación en XP se plantea como un constante diálogo entre el cliente y los desarrolladores, donde los primeros se encargarán de definir el alcance del proyecto, la prioridad de las HU, la composición de las versiones y la fecha de entrega de las mismas. Por otra parte, los desarrolladores estiman el tiempo necesario para implementar cada HU, el cual no debe ser mayor a tres semanas. Si la implementación excede las tres semanas, la HU debe ser modificada y dividida en otras más pequeñas. Además, en esta fase se generan artefactos de gran importancia como el Plan de Iteraciones y el Plan de Entregas.

### 2.6.1 Estimación de esfuerzo por HU

Estimación del esfuerzo por cada HU implicadas en el desarrollo de la aplicación:

**Tabla 5: Estimación de esfuerzos**

Número	Historias de Usuarios	Puntos de estimación
1	Autenticar usuario.	0.5
2	Gestionar datos del usuario.	1
3	Crear brigadas del SGOE.	0.5
4	Generar planificación del SGOE.	1
5	Generar PDF.	1
6	Mostrar planificación del SGOE.	0.5

7	Mostrar mi planificación del SGOE.	0.5
8	Mostrar reporte de cumplimiento del SGOE.	0.5
9	Controlar asistencia del SGOE.	0.5
10	Buscar integrantes en el SGOE para solicitar intercambio de turnos.	0.5
11	Intercambiar turnos en el SGOE.	0.5
12	Listar intercambio de turnos en el SGOE.	0.5
13	Gestionar rol.	0.5
14	Importar usuarios.	1
<b>Total</b>		9

### 2.6.2 Plan de entregas

*“El procedimiento Entrega proporciona el conjunto de actividades necesarias para llevar a cabo las entregas software y de documentación asociada a un proyecto, desde la formalización de la entrega hasta la revisión y validación de la misma.” (28)*

Debido a que el Plan de Entregas constituye un documento oficial debe ser lo más real posible. Por este plan los clientes le exigen a los desarrolladores la entrega de las distintas versiones del producto. El equipo de desarrollo de la propuesta de solución a partir de una reunión con el cliente, teniendo en cuenta las prioridades planteadas por el mismo y las estimaciones realizadas por los desarrolladores, elaboró el siguiente Plan de Entregas:

**Tabla 6: Plan de entregas**

Entregable	Final 1ra iteración	Final 2da iteración	Final 3ra iteración
	1ra semana de abril	3ra semana de abril	1ra semana de mayo
Sistema de planificación y control de la Guardia Obrera Estudiantil de la	Versión 0.1	Versión 0.2	Versión 1.0

<b>Facultad 4.</b>			
--------------------	--	--	--

**Tabla 7: Funcionalidades disponibles por entrega del producto**

Historia de Usuario	1ra entrega	2da entrega	3ra entrega
HU1. Autenticar usuario.	x	x	x
HU2. Gestionar datos del usuario.	x	x	x
HU3. Crear brigadas del SGOE.	x	x	x
HU4. Generar planificación del SGOE.	x	x	x
HU5. Generar PDF.	x	x	x
HU6. Mostrar planificación del SGOE.		x	x
HU7. Mostrar mi planificación del SGOE.		x	x
HU8. Mostrar reporte de cumplimiento del SGOE.		x	x
HU9. Mostrar asistencia del SGOE.		x	x
HU10. Buscar integrantes en el SGOE para solicitar intercambio de turnos.			x
HU11. Intercambiar turnos en el SGOE.			x
HU12. Listar intercambio de turnos en el SGOE.			x
HU13. Gestionar rol.			x
HU14. Importar usuarios.			x

### 2.6.3 Iteraciones

La metodología XP propone iteraciones de aproximadamente tres semanas de duración, definiéndose que HU se van a implementar en cada iteración. Las HU seleccionadas para la primera iteración deben abarcar los aspectos más importantes para la construcción de la estructura de todo el sistema. Además, en esta metodología una iteración no termina con la entrega del producto, por lo que una versión puede involucrar varias iteraciones del proceso de desarrollo.

El plan de duración de las iteraciones es el encargado de mostrar el orden de las HU que serán implementadas en cada una de las iteraciones. La elaboración de este plan contribuye a una mayor organización del trabajo. La siguiente tabla deja constancia del mismo.

**Tabla 8: Plan de duración de iteraciones**

Iteraciones	Orden de las HU a implementar	Duración de la iteración
<b>Iteración 1</b>	HU1. Autenticar usuario. HU2. Gestionar datos del usuario. HU3. Crear brigadas del SGOE. HU4. Generar planificación del SGOE. HU5. Generar PDF.	4 semanas
<b>Iteración 2</b>	HU6. Mostrar planificación del SGOE. HU7. Mostrar mi planificación del SGOE. HU8. Mostrar reporte de cumplimiento del SGOE. HU9. Controlar asistencia del SGOE.	2 semanas
<b>Iteración 3</b>	HU10. Buscar integrantes en el SGOE para solicitar intercambio de turnos. HU11. Intercambiar turnos en el SGOE. HU12. Listar intercambio de turnos en el SGOE. HU13. Gestionar rol. HU14. Importar usuarios.	3 semanas

## 2.7 Diseño

*“Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente.” (29)*

La metodología XP plantea que el diseño de la aplicación debe ser lo más sencilla posible, sin dejar de cumplir las expectativas del cliente. No se requieren los diagramas de clases para la descripción del sistema. XP hace uso de las tarjetas CRC (Contenido, Responsabilidad y Colaboración) para representar las clases.

### 2.7.1 Tarjetas CRC

Las tarjetas CRC debido a que son un modelo simple son fáciles de comprender y de aplicar. Estas recogen las responsabilidades de las clases. También representan las dependencias o colaboraciones de otras clases para realizar sus responsabilidades.

*“El objetivo de la misma es hacer, mediante tarjetas, un inventario de las clases que vamos a necesitar para implementar el sistema y la forma en que van a interactuar, de esta forma se pretende facilitar el análisis y discusión de las mismas por parte de varios actores del equipo de proyecto con el objeto de que el diseño sea lo más simple posible verificando las especificaciones del sistema.” (30)*

A continuación se muestran las tarjetas CRC generadas:

Tabla 9: Tarjeta CRC Asistencia

Tarjeta CRC	
<b>Clase:</b> Asistencia	
<b>Responsabilidades:</b> Es la clase encargada de guardar la asistencia de todos los usuarios del sistema al SGOE.	<b>Colaboraciones:</b> Usuario, Guardia, Reporte.

Tabla 10: Tarjeta CRC Brigada

Tarjeta CRC	
<b>Clase:</b> Brigada	
<b>Responsabilidades:</b> Clase que registra todos los integrantes de cada brigada.	<b>Colaboraciones:</b> Usuario, Guardia.

Tabla 11: Tarjeta CRC FosUser

Tarjeta CRC
-------------

<b>Clase:</b> FosUser	
<b>Responsabilidades:</b> Es la clase encargada de la autenticación en el sistema de los usuarios del dominio UCI.	<b>Colaboraciones:</b>

### 2.7.2 Patrones de diseño

El uso de frameworks para el desarrollo de software nos brinda la utilización de patrones de diseños ya incluidos en su arquitectura. Estos patrones son la base para la búsqueda de soluciones a problemas en el desarrollo de aplicaciones informáticas. Entre los patrones aplicables a diferentes situaciones se tienen: Patrones Generales de Software para Asignar Responsabilidades (en inglés *General Responsibility Assignment Software Patterns* (GRASP)) y Grupo de Cuatro (en inglés *Gang of Four* (GoF)).

#### 2.7.2.1 GRASP

*“Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones.”* (31)

Para lograr un diseño eficaz del software orientado a objetos se aplican los principios planteados por este patrón. A continuación se exponen los patrones GRASP utilizados para el diseño del sistema.

**Creador:** *“El patrón creador nos ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases. Éste patrón nos dice que la nueva instancia podrá ser creada por una clase si: contiene o agrega la clase, almacena la instancia en algún sitio (por ejemplo una base de datos) o usa directamente las instancias creadas del objeto.”* (32)

Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creadora. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. El mismo se evidencia en la clase Guardia1Controller.php cuando se crea un objeto de tipo brigada.

```
$t = $em->getRepository('GuardiaBundle:Brigada')->findAll();  
if (count($t) == 0) {  
    $brigada = new Brigada();  
    $brigada->setCompleta(true);  
    $em->persist($brigada);  
    $em->flush();  
}  
  
foreach ($estudiantes as $estudiante) {  
    $est = $em->getRepository('GuardiaBundle:Usuario')->findUsuariosByGrupo($estudiante->getGrupo());  
    $long = count($est);  
    $dif = $long % $cant;  
    if ($count < $cant and $estudiante->getGrupo() == $grupo and $dif == 0) {  
        $estudiante->setBrigada($brigada);  
        $em->persist($estudiante);  
    }  
  
    if ($count == $cant - 1) {  
        $brigada = new Brigada();  
        $brigada->setCompleta(true);  
    }  
}
```

Ilustración 2: Ejemplo patrón creador

**Bajo Acoplamiento:** “Es la idea de tener las clases lo menos ligadas entre sí que se pueda, de tal forma que, en caso de producirse una modificación en alguna de ellas, tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.” (32) La aplicación tiene una estructura que permite separar las clases pertenecientes al modelo de datos, las vistas de usuario y las clases controladoras.

**Experto:** “la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo o ejecutarlo, de este modo se obtendrá un diseño con mayor cohesión y cuya información se mantiene encapsulada, es decir, disminuye el acoplamiento.” (32) En la aplicación se evidencia este patrón en las entidades Asistencia.php, Brigada.php, FosUser.php, Intercambio.php, entre otras.

**Controlador:** “El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es el controlador quien recibe los datos del usuario y quien los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocio debe estar separada de la capa de presentación, lo que aumenta la reutilización de código y permite a la vez tener un mayor control.” (32) En la aplicación este patrón se evidencia en las clases controladoras AsistenciaController.php, FosUserController.php, GuardiaController.php, entre otras.

### 2.7.2.2 GoF

Basados en la experiencia acumulada por la resolución de problemas reiterativos son utilizados en situaciones frecuentes. Facilitan la construcción de clases reutilizables. Estos se clasifican en tres categorías basándose en su propósito: creacionales, estructurales y de comportamiento. A continuación se expone el patrón GoF utilizado para el diseño del sistema.

**Decorador:** *“permite añadir funcionalidades o responsabilidades a un objeto de forma dinámica. Este patrón provee una alternativa muy flexible a la creación de subclases para extender funcionalidades. Esto permitirá, en tiempo de ejecución, añadir esta funcionalidad o responsabilidad extra que necesita el objeto.”* (33)

Este patrón plantea la utilización de plantillas globales. Estas plantillas se encargarán de guardar el código que es común para varias plantillas del sistema, lo que garantiza que no se repita código en cada una de las interfaces. En la propuesta de solución a desarrollar se emplea una herencia de tres niveles para garantizar la máxima reutilización de código posible.



Ilustración 3: Ejemplo patrón decorador

## 2.8 Conclusiones del capítulo

Con el desarrollo del capítulo que finaliza se expusieron los artefactos generados por las dos primeras fases de la metodología XP, asegurando una correcta implementación por etapas. Además, se elaboraron

las HU que son la base para la construcción del sistema, estableciéndoles una prioridad teniendo en cuenta la importancia en la lógica del negocio y en que iteraciones deberán ser implementadas. Se construyó el Plan de entregas con las fechas en las que será liberada cada versión del sistema.

## Capítulo 3. Implementación y prueba.

La metodología XP plantea que la implementación de un producto software debe realizarse de forma iterativa e incremental, lo que va a permitir que se obtenga un producto funcional. Este entregable debe ser previamente probado para aumentar la visión de los desarrolladores y clientes ante potenciales cambios. En el presente capítulo se define la arquitectura de la propuesta de solución y se analizan los estándares de codificación. Por último, se realizan las pruebas para detectar y corregir posibles errores.

### 3.1 Arquitectura

El uso del framework Symfony 2 determinó la arquitectura de la propuesta de solución. Symfony 2 permite el desarrollo de aplicaciones web que empleen una arquitectura Modelo-Vista-Controlador. Este patrón arquitectónico plantea la separación del código en tres capas.

*“El modelo representa la parte de la aplicación que implementa la lógica de negocio. Esto significa que es responsable de la recuperación de datos convirtiéndolos en conceptos significativos para la aplicación, así como su procesamiento, validación, asociación y cualquier otra tarea relativa a la manipulación de dichos datos.” (34)*

Los objetos del modelo representan los principales conceptos en torno a los cuales se desea implementar un programa. En esta capa actúan los ORM Propel y Doctrine. Esto le permite al desarrollador una mayor abstracción de la base de datos.

Las vistas contienen el código de la aplicación que va a producir la visualización de las interfaces de usuarios. Estas hacen una presentación de los datos del modelo al estar separadas de los objetos del mismo. La Vista es principalmente la capa de plantillas PHP con las que el usuario interactúa.

*“En realidad es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo.” (35)*

La capa del Controlador gestiona las peticiones de los usuarios. Esta capa es la pieza del código que llama al modelo para obtener algunos datos que le pasa a la Vista para la presentación al cliente. Esto convierte al Controlador en el responsable de responder a la información solicitada.

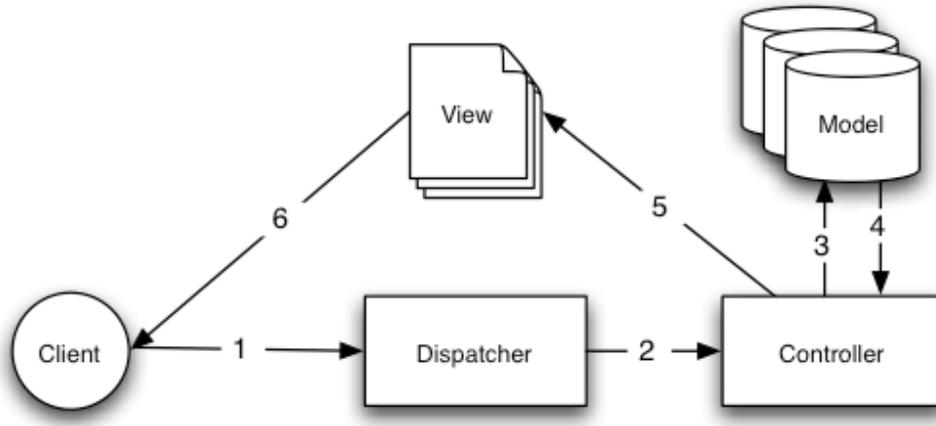


Ilustración 4: Petición MVC

### 3.2 Estándar de codificación

*“La codificación debe hacerse atendiendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.”* (36) La metodología XP propone el uso de estándares de codificación. Esto se refuerza con la programación en parejas, la propiedad colectiva del código y la integración continua. De esta forma el código no es conocido por una sola persona, esto permite que otros miembros del equipo puedan realizar cambios en el mismo.

A continuación se muestran los principales estándares de codificación utilizados en la implementación del sistema:

- Añadir un solo espacio después de cada delimitador (coma).
- Añadir un solo espacio antes y después de los operadores (==, and, or y otros).
- Añadir una coma después de cada elemento del arreglo en un arreglo multilíneas.
- Los comentarios multilíneas en el código PHP se escriben al iniciar con los caracteres `“/*”` y al terminar con `“*/”`, los comentarios de una sola línea comienzan con los caracteres `“//”`.
- El código se encuentra tabulado a través del formato que aplica la combinación de teclas ALT+SHIFT+F del NetBeans IDE.

- Usar llaves para indicar la estructura del cuerpo de control, independientemente del número de declaraciones que contenga.
- Declarar las propiedades de cada clase antes que los métodos. Utilizar paréntesis al crear instancias de clases, independientemente del número de argumentos que el constructor tenga.
- Los nombres de las clases del modelo utilizan el estándar *Upper Camel Case* y las clases controladoras y los métodos utilizan en sus nombres el estándar *Camel Case*.
- Todos los atributos de las clases del modelo están encapsulados. Son privados y el acceso a ellos se hace por medio de los métodos `getAtributo` y/o `setAtributo`.
- Las variables deben ser explícitas, aunque se pueden usar abreviaturas siempre y cuando no violen este principio.

El uso de estos estándares de codificación permite el trabajo coordinado de los desarrolladores, garantiza una mejor comprensión del código en la implementación y en las fases futuras de mantenimiento. Además, el código ganará en limpieza, claridad y organización.

### 3.3 Bundles del sistema

*“Los bundles son la parte más importante de Symfony2. Permiten utilizar funcionalidades construidas por terceros o empaquetar tus propias funcionalidades para distribuirlas y reutilizarlas en otros proyectos. Además, facilitan mucho la activación o desactivación de determinadas características dentro de una aplicación.” (37)*

A continuación se muestran los bundles usados en la propuesta de solución:

- `FOSUserBundle`: encargado de gestionar la autenticación de los usuarios al sistema.
- `SnappyBundle`: utiliza la librería `snappy` que permite generar PDF a partir de una página HTML.
- `SuperAdminBundle`: se encarga de gestionar las funcionalidades del rol: `ROL_SUPER_ADMIN`.
- `AdminBundle`: se encarga de gestionar las funcionalidades del rol: `ROL_ADMIN`.
- `UsuarioBundle`: se encarga de gestionar las funcionalidades del rol: `ROL_USER`.
- `LiuggioExcelBundle`: permite el trabajo con la librería `PHPExcel`.

### 3.4 Tareas de ingeniería

*“Se desarrolla en cada iteración el conjunto de historias de usuario que se han seleccionado para la misma. Cada una de ellas se clasifica en tareas de ingeniería que vendrían a considerarse como las entradas de trabajo para cada equipo de programadores. A nivel documental, como sucede con las historias de usuario, las tareas de ingeniería vendrían a ser como una ficha que contendría el número identificador de la tarea, el identificador de la historia de usuario con la que está relacionada, el nombre de la tarea, el tipo (nuevo desarrollo, corrección, mejora, etc...), la fecha de inicio, su fecha de fin, el equipo responsable y la descripción.” (38)*

La metodología XP plantea dividir las HU en Tareas de Ingeniería (TI) con el propósito de facilitar la implementación del sistema a desarrollar. Para lograr obtener un resultado satisfactorio una HU se puede desglosar en varias TI. Las HU generadas fueron divididas en 18 TI, de las cuales se muestran tres, correspondientes a las HU 1, HU 2 y HU 3. Las restantes se encuentran en los anexos.

**Tabla 12: Tarea de ingeniería 1**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 1	<b>No. De la HU:</b> 1
<b>Nombre de la tarea:</b> Autenticar usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 2/03/2015	<b>Fecha fin:</b> 4/03/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Modificar y hacer uso de las clases webserviceusuarios, webServiceUser y webserviceuserprovider, e implementar la clase FosUser para garantizar la autenticación de los usuarios en el sistema.	

**Tabla 13: Tarea de ingeniería 2**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 2	<b>No. De la HU:</b> 2
<b>Nombre de la tarea:</b> Actualizar datos del usuario.	

<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 7
<b>Fecha inicio:</b> 5/03/2015	<b>Fecha fin:</b> 12/03/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Con el empleo del código php app/console doctrine:generate:crud, generar el crud necesario para que el administrador pueda modificar la información del usuario	

Tabla 14: tarea de Ingeniería 6

Tarea de Ingeniería	
<b>No. De la tarea:</b> 5	<b>No. De la HU:</b> 3
<b>Nombre de la tarea:</b> Crear brigada.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 13/03/2015	<b>Fecha fin:</b> 17/03/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Implementar el método crearBrigadaAction() que se encargará de asignar los estudiantes a las brigadas del servicio de guardia.	

### 3.5 Funcionamiento del sistema

El sistema está compuesto por 4 módulos: Roles y Usuarios, Gestionar Guardia, Control y Reportes y Opciones de Usuario. La implementación de estos módulos es la respuesta a la propuesta de los procesos automatizables definidos para el Sistema de planificación y control de la Guardia Obrera-Estudiantil de la Facultad 4. Su implementación se realizó dando cumplimiento a las tareas de ingeniería descritas anteriormente. A continuación se detalla el funcionamiento de los módulos mencionados anteriormente.

**Roles y Usuarios:** Gestiona la información de los roles definidos en el sistema. Brinda la posibilidad de modificar el rol de los usuarios locales y de los usuarios del dominio uci. Los usuarios registrados y autenticados en el sistema inicialmente poseen el rol usuario, por lo que este módulo permite asignarles los roles Super-administrador y administrador. Además, gestiona toda la información de los usuarios que se encuentran registrados en el sistema. Permite modificar los datos (nombre, edad, grupo, sexo, categoría y disponibilidad) de los mismos y posibilita insertar un usuario nuevo en el sistema.

**Gestionar Guardia:** Gestiona toda la información referente al SGOE. Este módulo permite crear brigadas, planificar la guardia de los estudiantes y de los profesores, así como mostrar la planificación realizada.

**Control y Reportes:** Gestiona toda la información relacionada con la asistencia de los usuarios al SGOE. Este módulo permite modificar y mostrar la asistencia de los usuarios, así como descargar los reportes de esta información.

**Opciones de usuario:** Este módulo contiene las opciones que el sistema brinda a los usuarios, las cuales son: Mis asistencias (muestra la asistencia del usuario autenticado en la aplicación), Mi planificación (muestra la planificación del usuario autenticado en la aplicación), Seleccionar Usuario (posibilita buscar otro usuario y ver su planificación), Solicitar Intercambio (permite realizar la solicitud de intercambio con otro usuario) y Listado de Intercambio (muestra la solicitud de intercambio y la opción de aceptarla o denegarla).

### 3.6 Pruebas

*“El único instrumento adecuado para determinar el status de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos.” (39)*

Las pruebas de software son una actividad más en el proceso de control de calidad. El objetivo principal de las mismas es descubrir los errores existentes en el software con la menor cantidad de esfuerzo y de tiempo. Entre las diferentes técnicas para la ejecución de las pruebas se encuentran las pruebas unitarias y las pruebas de aceptación.

#### 3.6.1 Pruebas unitarias

*“Este tipo de pruebas son ejecutadas normalmente por el equipo de desarrollo, básicamente consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada.” (40)*

Las pruebas unitarias brindan la posibilidad de comprobar el correcto funcionamiento de un módulo de código. Symfony 2 facilita la realización de los test (pruebas) unitarios con el uso de la librería PHPUnit. A través de este framework de desarrollo se puede verificar que el comportamiento de una unidad de código

es el esperado. Cuando ocurre un error, PHPUnit muestra el mensaje "Failure" (fallo) y muestra el listado de todos los tests que han fallado. Cuando todos los tests se han ejecutado correctamente se muestra el mensaje "Ok" (bien).

Las siguientes imágenes representan las pruebas realizadas a la clase Guardia.php.

```
Configuration read from /var/www/html/Symfony (2)/app/phpunit.xml.dist
F.
Time: 1.06 seconds, Memory: 14.50Mb
There was 1 failure:
1) Guardia\GuardiaBundle\Tests\Controller\GuardiaControllerTest::RutaGuardia
Las ruta guardia no se encuentra.
Failed asserting that 404 matches expected 302.
/var/www/html/Symfony (2)/src/Guardia/GuardiaBundle/Tests/Controller/GuardiaControllerTest.php:22
FAILURES!
Tests: 2, Assertions: 2, Failures: 1.
```

Ilustración 4: Prueba unitaria a la entidad Guardia

```
Configuration read from /var/www/html/Symfony (2)/app/phpunit.xml.dist
..
Time: 502 ms, Memory: 10.50Mb
OK (2 tests, 2 assertions)
```

Ilustración 5: Prueba unitaria a la entidad Guardia

Fueron realizadas dos iteraciones de pruebas Unitarias. Errores encontrados:

1ra iteración:

- Rutas no asignadas.
- Validación de campos obligatorios.

Los errores fueron corregidos y en la 2da iteración no se detectaron errores.

### 3.6.2 Pruebas de aceptación

*“El uso de cualquier producto de software tiene que estar justificado por las ventajas que ofrece. Sin embargo, antes de empezar a usarlo es muy difícil determinar si sus ventajas realmente justifican su uso. El mejor instrumento para esta determinación es la llamada 'prueba de aceptación'. En esta prueba se evalúa el grado de calidad del software con relación a todos los aspectos relevantes para que el uso del producto se justifique.” (41)*

Las pruebas de aceptación, por cada TI, describen como el sistema se comporta en cierto tipo de situaciones y que resultados arroja. Las conclusiones obtenidas de estas pruebas deben ser del conocimiento de todo el equipo de desarrollo. A continuación se muestran los tres primeros casos de prueba de aceptación, el resto se encuentra en los anexos.

**Tabla 15: Prueba de aceptación No. 1**

Pruebas de aceptación	
<b>Número:</b> 1	<b>No. HU:</b> 1
<b>Nombre:</b> Autenticar usuario.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario autenticarse.	
<b>Condiciones de ejecución:</b> Verificar que el usuario este previamente registrado en el sistema.	
<b>Pasos de ejecución:</b> El usuario ingresa su usuario y contraseña, el sistema valida que los datos sean correctos y se muestra la interfaz principal de la aplicación.	
<b>Resultado esperado:</b> El usuario se autentica satisfactoriamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 16: Prueba de aceptación No. 2**

Pruebas de aceptación	
<b>Número:</b> 2	<b>No. HU:</b> 2
<b>Nombre:</b> Modificar datos del usuario.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario actualizar los datos del	

usuario.
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Super-administrador y debe existir al menos un usuario en la base de datos del sistema.
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Gestionar usuarios" en el menú principal. Luego el sistema brinda la opción "Modificar", permitiéndole cambiar la información del usuario.
<b>Resultado esperado:</b> Se registran en la base de datos del sistema los datos de los usuarios.
<b>Evaluación de la prueba:</b> Satisfactoria.

Tabla 17: Prueba de aceptación No. 3

Pruebas de aceptación	
<b>Número:</b> 3	<b>No. HU:</b> 2
<b>Nombre:</b> Insertar usuario.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario insertar otro usuario al sistema.	
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Administrador.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Gestionar usuarios" en el menú principal. Luego selecciona la opción "Crear nuevo usuario" y el sistema mostrará un formulario en el que se especificará los datos del usuario. Finalmente el usuario selecciona la opción "Insertar" y el sistema añade al nuevo usuario.	
<b>Resultado esperado:</b> El sistema inserta un nuevo usuario del SGOE.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

### 3.6.3 Resultados de las pruebas

Se definieron 19 casos de pruebas de aceptación, uno por cada TI elaborada. Luego de finalizadas las pruebas de aceptación se detectaron varias no conformidades. Las no conformidades encontradas fueron resueltas en cada iteración antes de pasar a la siguiente. La Tabla 69 muestra el resultado de las pruebas.

Tabla 18: Resultado pruebas de aceptación.

Resultado pruebas de aceptación
---------------------------------

<b>Iteración</b>	Iteración 1	Iteración 2	Iteración 3
<b>Cantidad de errores</b>	19	12	5

### **3.7 Conclusiones del capítulo**

Con la realización del capítulo que concluye se generaron las tarjetas CRC, artefacto necesario para la implementación de la propuesta de solución. Estas permitieron identificar las clases del sistema a desarrollar y la relación entre ellas. Un mejor entendimiento del código fue posible gracias a los estándares de codificación. Estos facilitan una posterior modificación y mantenimiento del código fuente de la aplicación. Las tareas de ingeniería permitieron una correcta implementación de las funcionalidades descritas en las historias de usuarios. Por último las pruebas unitarias y de aceptación realizadas permitieron corregir los errores y obtener un software con la calidad requerida.

## **Conclusiones Generales**

Con el desarrollo de la presente investigación y realizada la propuesta de solución se puede afirmar que se cumplieron los objetivos propuestos, lo que permite arribar a las siguientes conclusiones:

- El análisis de la bibliografía consultada permitió, mediante el empleo de diferentes métodos científicos, conocer de la existencia de diferentes sistemas que permiten la planificación del Servicio de Guardia Obrera-Estudiantil.
- Los sistemas similares estudiados no cumplen con los requerimientos necesarios para realizar la planificación de la Guardia Obrera-Estudiantil en la Facultad 4. Por tal motivo se definió una propuesta de solución que se adapta a la estructura actual del Servicio de Guardia Obrera-Estudiantil.
- La aplicación diseñada permite la gestión del Servicio de Guardia Obrera-Estudiantil, mediante asignación de zonas, postas y turnos a los usuarios insertados en el sistema, teniendo en cuenta el modelo de guardia aplicado actualmente en la Universidad de las Ciencias Informáticas.
- Se realizó la implementación del Sistema de planificación y control de la Guardia Obrera-Estudiantil, teniendo en cuenta las herramientas y metodologías de desarrollo seleccionadas.
- Las pruebas unitarias y de aceptación aplicadas al sistema desarrollado permitieron validar que el software elaborado cumpliera con los requisitos funcionales planteados por el cliente.

## Recomendaciones

Tomando como base la investigación realizada y el análisis de los resultados obtenidos se recomienda:

- Realizar el despliegue del sistema después una vez aprobada, por Seguridad Informática, la propuesta realizada por los autores de la presente investigación.
- Añadir al sistema una leyenda que muestre aspectos fundamentales del servicio de guardia, para facilitar la gestión de las postas y zonas.
- Se recomienda, para futuras versiones, el empleo de la clase SwapCliente para acceder a los métodos que definen los Servicios Web UCI y obtener la información necesaria de los usuarios del SGOE a insertar en la base de datos del sistema.

- **Referencias Bibliográficas**

1. **Diccionario de la lengua española | Real Academia Española.** [en línea]. [Accedido 7 marzo 2015]. Disponible desde: <http://lema.rae.es/drae/?d=drae&val=planificar&x=0&y=0#>.
2. **Diccionario de la lengua española | Real Academia Española.** [en línea]. [Accedido 7 marzo 2015]. Disponible desde: <http://lema.rae.es/drae/?d=drae&val=turno&x=0&y=0>
3. **Diccionario de la lengua española | Real Academia Española.** [en línea]. [Accedido 7 marzo 2015]. Disponible desde: <http://lema.rae.es/drae/?d=drae&val=zona&x=0&y=0>
4. **Diccionario de la lengua española | Real Academia Española.** . [en línea]. [Accedido 7 marzo 2015]. Disponible desde: <http://lema.rae.es/drae/?d=drae&val=brigada&x=0&y=0>
5. **Diccionario de la lengua española | Real Academia Española.** . [en línea]. [Accedido 7 marzo 2015]. Disponible desde: <http://lema.rae.es/drae/?d=drae&val=asistencia&x=0&y=0>
6. **Diccionario de la lengua española | Real Academia Española.** . [en línea]. [Accedido 7 marzo 2015]. Disponible desde: <http://lema.rae.es/drae/?d=drae&val=reporte&x=0&y=0>
7. **TURNEX: Excelencia en la Gestión Operativa de Turnos.** [en línea]. [Accedido 8 febrero 2015]. Disponible desde: <http://solexservices.com/index.php/noticias/110-turnex-excelencia-en-la-gestion-operativa-de-turnos>
8. **Metodología Híbrida para Desarrollo de Software en México.** Jiménez-Hernández, Eréndira M. 2012. Ciudad de México: s.n., 2012.
9. **Qué es SCRUM | proyectos Ágiles.** [en línea]. [Accedido 5 febrero 2015]. Disponible desde: <http://www.proyectosagiles.org/que-es-scrum>
10. **Pressman, Roger S.** 2010. Software Engineering a Practitioner's Approach. 2010. Seventh Edition.
11. **Mora, Sergio Luján.** 2002. Programación de Aplicaciones Web: Historia, Principios Básicos y Clientes Web. Alicante: Club Universitario. 2002.
12. **Gutiérrez, Demián.** 2010. Framework y Componentes. S.1.: Universidad de los Andes, 2010.
13. **Technologies, Z. 2009. Programmer's Reference Guide, Zend Framework.** [en línea]. [Accedido: 5 febrero 2015]. Disponible desde: <http://framework.zend.com/manual/en/index.html>

14. **¿Qué es Symfony?** [en línea]. [Accedido 8 febrero 2015]. Disponible desde: <http://symfony.es/que-es-symfony>
15. **¿Qué es Bootstrap y cómo funciona en el diseño web?** [en línea]. [Accedido 8 febrero 2015]. Disponible desde: <http://www.arweb.com/chucherias/editorial/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web.htm>
16. **Introducción a PHP – Manual de PHP.** [en línea]. [Accedido 5 febrero 2015]. Disponible desde: <http://www.manualdephp.com/manualdephp/introduccion-a-php/>
17. **Notas.** [en línea]. [Accedido 5 febrero 2015]. Disponible desde: <http://www.uv.es/jac/guia/notas.htm>
18. **HTML5 – HTML | MDN.** [en línea]. [Accedido 5 febrero 2015]. Disponible desde: <https://developer.mozilla.org/es/docs/HTML/HTML5>
19. **Capítulo 1. Introducción (Introducción a CSS).** [en línea]. [Accedido 5 febrero 2015]. Disponible desde: [http://librosweb.es/css/capitulo\\_1.html](http://librosweb.es/css/capitulo_1.html)
20. **Capítulo 1. Introducción (Introducción a JavaScript).** [en línea]. [Accedido 5 febrero 2015]. Disponible desde: [http://librosweb.es/javascript/capitulo\\_1.html](http://librosweb.es/javascript/capitulo_1.html)
21. **PostgreSQL: Características, limitaciones y ventajas.** [en línea]. [Accedido 8 febrero 2015]. Disponible desde: <http://postgresql-dbms.blogspot.com/p/limitaciones-puntos-de-recuperacion.html>
22. **¿Qué es MySQL?: esepestudio, especialistas web.** [en línea]. [Accedido 8 febrero 2015]. Disponible desde: <http://www.esepestudio.com/noticias/que-es-mysql>
23. **Apache - openSUSE.** [en línea]. [Accedido 8 febrero 2015]. Disponible desde: <https://es.opensuse.org/Apache>
24. **Capítulo 2. Introducción a UML.** [en línea]. [Accedido 7 febrero 2015]. Disponible desde: <https://docs.kde..org/stable/es/kdesdk/umbrello/uml-basics.html>
25. **Requerimientos Funcionales y No Funcionales (RF/RNF).** [en línea]. Accedido 26 marzo 2015. Disponible desde: <http://ingenieriadesoftware.bligoo.com.mx/requerimientos-funcionales-y-no-funcionales-rf-rnf>

26. **2.3 METODOLOGIAS.** [en línea]. [Accedido 31 Marzo 2015]. Disponible desde: <https://grupo13sistemaacademicoescolar.wordpress.com/2-3-metodologias/>
27. **Ciclo de vida de un proyecto XP.** [en línea]. [Accedido 31 Marzo 2015]. Disponible desde: <http://oness.sourceforge.net/proyecto/html/ch05s02.html>
28. **Procedimiento de Entrega de Software | Marco de Desarrollo de la Junta de Andalucía.** [en línea]. [Accedido 31 Marzo 2015]. Disponible desde: <http://www.juntadeandalucia.es/servicios/madeja/contenido/procedimiento/10>
29. **CyTA.** [en línea]. [Accedido 19 mayo 2015]. Disponible desde: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
30. **Desarrollo de software. Tarjetas CRC | Jummp.** [en línea]. [Accedido 4 mayo 2015]. Disponible desde: <https://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>
31. **Patrones GRASP | Prácticas de Software.** [en línea]. [Accedido 8 mayo 2015]. Disponible desde: <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp>
32. **SOLID y GRASP - Buenas prácticas hacia el éxito en el desarrollo de software.** [en línea]. [Accedido 8 mayo 2015]. Disponible desde: <http://collection.openlibra.com.s3.amazonaws.com/pdf/solid-y-grasp-buenas-practicas-hacia-el-exito-en-el-desarrollo-de-software.pdf>
33. **Patrones de diseño (GoF) | Mundo Informático.** [en línea]. [Accedido 9 mayo 2015]. Disponible desde: <https://infow.wordpress.com/category/patrones-de-disenogof/page/2/>
34. **Entendiendo el Modelo - Vista - Controlador — documentación de CakePHP Cookbook - 2.x.** [en línea]. [Accedido 4 mayo 2015]. Disponible desde: <http://book.cakephp.org/2.0/es/cakephp-overview/understanding-model-view-controller.html>
35. **Qué es MVC.** [en línea]. [Accedido 4 mayo 2015]. Disponible desde: <http://www.desarrolloweb.com/articulos/que-es-mvc.html>
36. **Fases.** [en línea]. [Accedido 13 mayo 2015]. Disponible desde: <http://programacionextrema.tripod.com/fases.html#terceraFase>
37. **4.4 El sistema de bundles (Symfony 2.4, el libro oficial).** [en línea]. [Accedido 13 mayo 2015]. Disponible desde: [http://librosweb.es/libro/symfony\\_2\\_x/capitulo\\_4/el\\_sistema\\_de\\_bundles.html](http://librosweb.es/libro/symfony_2_x/capitulo_4/el_sistema_de_bundles.html)

38. **Desarrollo de software. Programación extrema (eXtreme Programming: XP) II | Jummp.** [en línea]. [Accedido 19 mayo 2015]. Disponible desde: <https://jummp.wordpress.com/2011/04/27/desarrollo-de-software-programacion-extrema-extreme-programming-xp-ii/>
39. **Las pruebas de software forman el único instrumento adecuado para determinar la calidad de software.** [en línea]. [Accedido 26 mayo 2015]. Disponible desde: <http://pruebasdesoftware.com/laspruebasdesoftware.htm>
40. **PRUEBAS DEL SOFTWARE | Ingeniería de Software con énfasis en pruebas.** [en línea]. [Accedido 26 mayo 2015]. Disponible desde: <https://pruebasdelsoftware.wordpress.com/>
41. **La prueba de aceptación es la prueba más importante para los productos software.** [en línea]. [Accedido 26 mayo 2015]. Disponible desde: <http://pruebasdesoftware.com/pruebadeaceptacion.htm>

## Anexos

## Anexo 1: Historias de Usuario

Tabla 19: Historia de Usuario No. 3

Historia de usuario	
<b>Número:</b> 3	<b>Usuario:</b> Administrador.
<b>Nombre:</b> Crear brigadas del SGOE.	
<b>Prioridad en el negocio:</b> Alta.	<b>Riesgo de desarrollo:</b> Alto.
<b>Puntos de estimación:</b> 0.5	<b>Iteración:</b> 1
<b>Descripción:</b> El sistema brindará la posibilidad al usuario Administrador de crear las brigadas, las mismas estarán compuestas por no menos de 4 estudiantes.	

Tabla 20: Historia de Usuario No. 4

Historia de usuario	
<b>Número:</b> 4	<b>Usuario:</b> Administrador.
<b>Nombre:</b> Generar planificación del SGOE.	
<b>Prioridad en el negocio:</b> Alta.	<b>Riesgo de desarrollo:</b> Alto.
<b>Puntos de estimación:</b> 1	<b>Iteración:</b> 1
<b>Descripción:</b> El sistema tendrá la posibilidad de generar automáticamente la planificación de la Guardia Obrera-Estudiantil en la Facultad 4. Teniendo en cuenta: la categoría docente, el sexo (las mujeres no pueden hacer el último turno de guardia en ninguna de las zonas), si reside en la universidad y si está disponible.	

Tabla 21: Historia de Usuario No. 5

Historia de usuario	
<b>Número:</b> 5	<b>Usuario:</b> Administrador.
<b>Nombre:</b> Generar PDF.	
<b>Prioridad en el negocio:</b> Baja.	<b>Riesgo de desarrollo:</b> Media.
<b>Puntos de estimación:</b> 1	<b>Iteración:</b> 1

**Descripción:** El sistema tendrá la posibilidad de exportar en formato PDF la información almacenada en el sistema.

**Tabla 22: Historia de Usuario No. 6**

Historia de usuario	
<b>Número:</b> 6	<b>Usuario:</b> Administrador.
<b>Nombre:</b> Mostrar planificación del SGOE.	
<b>Prioridad en el negocio:</b> Media.	<b>Riesgo de desarrollo:</b> Medio.
<b>Puntos de estimación:</b> 0.5	<b>Iteración:</b> 2
<b>Descripción:</b> El sistema tendrá la posibilidad de mostrar la planificación del SGOE realizada.	

**Tabla 23: Historia de Usuario No. 7**

Historia de usuario	
<b>Número:</b> 7	<b>Usuario:</b> Estándar.
<b>Nombre:</b> Mostrar mi planificación del SGOE.	
<b>Prioridad en el negocio:</b> Baja.	<b>Riesgo de desarrollo:</b> Baja.
<b>Puntos de estimación:</b> 0.5	<b>Iteración:</b> 2
<b>Descripción:</b> El sistema tendrá la posibilidad de mostrar la planificación del SGOE realizada para un usuario específico.	

**Tabla 24: Historia de Usuario No. 8**

Historia de usuario	
<b>Número:</b> 8	<b>Usuario:</b> Administrador.
<b>Nombre:</b> Mostrar reportes de cumplimientos del SGOE.	
<b>Prioridad en el negocio:</b> Baja.	<b>Riesgo de desarrollo:</b> Baja.
<b>Puntos de estimación:</b> 0.5	<b>Iteración:</b> 2
<b>Descripción:</b> El sistema tendrá la posibilidad de visualizar la información del cumplimiento del SGOE.	

Tabla 25: Historia de Usuario No. 9

Historia de usuario	
<b>Número:</b> 9	<b>Usuario:</b> Administrador, Estándar.
<b>Nombre:</b> Controlar asistencia del SGOE.	
<b>Prioridad en el negocio:</b> Media.	<b>Riesgo de desarrollo:</b> Bajo.
<b>Puntos de estimación:</b> 0.5	<b>Iteración:</b> 2
<b>Descripción:</b> El sistema tendrá la posibilidad de controlar la asistencia de los estudiantes y profesores al servicio de guardia.	

Tabla 26: Historia de Usuario No. 10

Historia de usuario	
<b>Número:</b> 10	<b>Usuario:</b> Estándar.
<b>Nombre:</b> Buscar integrantes en el SGOE para solicitar intercambio.	
<b>Prioridad en el negocio:</b> Media.	<b>Riesgo de desarrollo:</b> Medio.
<b>Puntos de estimación:</b> 0.5	<b>Iteración:</b> 3
<b>Descripción:</b> El sistema brindara a los usuarios la opción de búsqueda de otras personas y solicitar intercambio del servicio de guardia.	

Tabla 27: Historia de Usuario No. 11

Historia de usuario	
<b>Número:</b> 11	<b>Usuario:</b> Estándar.
<b>Nombre:</b> Intercambiar turnos en el SGOE.	
<b>Prioridad en el negocio:</b> Media.	<b>Riesgo de desarrollo:</b> Medio.
<b>Puntos de estimación:</b> 1	<b>Iteración:</b> 3
<b>Descripción:</b> El sistema tendrá la opción de realizar el intercambio de turnos del SGOE entre los usuarios.	

Tabla 28: Historia de Usuario No. 12

Historia de usuario	
<b>Número:</b> 12	<b>Usuario:</b> Administrador.
<b>Nombre:</b> Listar intercambios de turnos en el SGOE.	
<b>Prioridad en el negocio:</b> Media.	<b>Riesgo de desarrollo:</b> Medio.
<b>Puntos de estimación:</b> 0.5	<b>Iteración:</b> 3
<b>Descripción:</b> El sistema tendrá la posibilidad de mostrar las solicitudes de intercambio que tiene un usuario y podrá aceptarlas o rechazarlas.	

Tabla 29: Historia de Usuario No. 13

Historia de usuario	
<b>Número:</b> 13	<b>Usuario:</b> Super-Administrador.
<b>Nombre:</b> Gestionar rol.	
<b>Prioridad en el negocio:</b> Media.	<b>Riesgo de desarrollo:</b> Medio.
<b>Puntos de estimación:</b> 0.5	<b>Iteración:</b> 3
<b>Descripción:</b> El sistema brindará al Super-administrador poder definir los roles de los usuarios en el sistema.	

Tabla 30: Historia de Usuario No. 14

Historia de usuario	
<b>Número:</b> 14	<b>Usuario:</b> Super-Administrador.
<b>Nombre:</b> Importar usuarios.	
<b>Prioridad en el negocio:</b> Media.	<b>Riesgo de desarrollo:</b> Medio.
<b>Puntos de estimación:</b> 1	<b>Iteración:</b> 3
<b>Descripción:</b> El sistema brindará al Super-administrador la posibilidad de añadir a la base de datos del sistema todos los usuarios del servicio de guardia.	

## Anexo 2: Tarjetas CRC

Tabla 31: Tarjeta CRC Guardia

<b>Tarjeta CRC</b>	
<b>Clase:</b> Guardia	
<b>Responsabilidades:</b> Es la clase encargada de gestionar todo lo referente al SG (planificación, modificación y control).	<b>Colaboraciones:</b> Brigada, Usuario.

Tabla 32: Tarjeta CRC Intercambio

<b>Tarjeta CRC</b>	
<b>Clase:</b> Intercambio	
<b>Responsabilidades:</b> Es la clase encargada de registrar los intercambios de turnos del SGOE realizados por dos usuarios.	<b>Colaboraciones:</b> Usuario.

Tabla 33: Tarjeta CRC Reporte

<b>Tarjeta CRC</b>	
<b>Clase:</b> Reporte	
<b>Responsabilidades:</b> Es la clase encargada de guardar la situación existente y registrada del SG en cuanto a cumplimiento, además de generar documentos en formato PDF.	<b>Colaboraciones:</b> Asistencia.

Tabla 34: Tarjeta CRC Usuario

<b>Tarjeta CRC</b>	
<b>Clase:</b> Usuario	
<b>Responsabilidades:</b> Es la clase encargada de gestionar todo lo referente a los usuarios (estudiantes y profesores).	<b>Colaboraciones:</b> Asistencia, Brigada, Intercambio, Reporte, Guardia.

Tabla 35: Tarjeta CRC Rol

<b>Tarjeta CRC</b>
--------------------

<b>Clase:</b> Rol	
<b>Responsabilidades:</b> Es la clase encargada de definir los roles de los usuarios en el sistema (Super_administrador, Administrador y Estándar).	<b>Colaboraciones:</b> FosUser

**Tabla 36: Tarjeta CRC WebServiceUsuarios**

<b>Tarjeta CRC</b>	
<b>Clase:</b> WebServiceUsuarios	
<b>Responsabilidades:</b> Es la clase encargada de gestionar los roles de los usuarios del dominio uci.	<b>Colaboraciones:</b>

**Tabla 37: Tarjeta CRC WebServiceUser**

<b>Tarjeta CRC</b>	
<b>Clase:</b> WebserviceUser	
<b>Responsabilidades:</b> Es la clase encargada de la autenticación de los usuarios del dominio uci.	<b>Colaboraciones:</b>

**Tabla 38: Tarjeta CRC WebserviceUserProvider**

<b>Tarjeta CRC</b>	
<b>Clase:</b> WebserviceUserProvider	
<b>Responsabilidades:</b> Es la clase encargada de realizar la autenticación en el sistema.	<b>Colaboraciones:</b>

**Anexo 3: Tareas de Ingeniería**

**Tabla 39: Tarea de ingeniería 3**

<b>Tarea de Ingeniería</b>	
<b>No. De la tarea:</b> 3	<b>No. De la HU:</b> 2
<b>Nombre de la tarea:</b> Insertar usuario.	

<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 7
<b>Fecha inicio:</b> 5/03/2015	<b>Fecha fin:</b> 12/03/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Con el empleo del código php app/console doctrine:generate:crud, generar el crud necesario para que el administrador pueda insertar un nuevo usuario.	

**Tabla 40: Tarea de ingeniería 4**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 4	<b>No. De la HU:</b> 2
<b>Nombre de la tarea:</b> Mostrar usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 7
<b>Fecha inicio:</b> 5/03/2015	<b>Fecha fin:</b> 12/03/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Con el empleo del código php app/console doctrine:generate:crud, generar el crud necesario para que el administrador pueda visualizar todos los usuarios.	

**Tabla 41: Tarea de ingeniería 6**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 6	<b>No. De la HU:</b> 4
<b>Nombre de la tarea:</b> Generar planificación de estudiantes.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 7
<b>Fecha inicio:</b> 18/03/2015	<b>Fecha fin:</b> 25/03/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Implementar el método planificarEstudiantesAction() responsable de planificar el servicio de guardia para los estudiantes.	

**Tabla 42: Tarea de ingeniería 7**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 7	<b>No. De la HU:</b> 4
<b>Nombre de la tarea:</b> Generar planificación de profesores.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 7
<b>Fecha inicio:</b> 18/03/2015	<b>Fecha fin:</b> 25/03/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Implementar el método planificarProfesoresAction() responsable de planificar el servicio de guardia para los profesores.	

Tabla 43: Tarea de ingeniería 8

Tarea de Ingeniería	
<b>No. De la tarea:</b> 8	<b>No. De la HU:</b> 5
<b>Nombre de la tarea:</b> Generar PDF.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 7
<b>Fecha inicio:</b> 25/03/2015	<b>Fecha fin:</b> 1/04/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Utilizar el bundle SnappyBundle e implementar los métodos GenerarTodasAsistenciasPDFAction(), GenerarEstudiantesAsistenciasPDFAction(), GenerarUltimaAsistenciasEstudiantesPDFAction(), GenerarProfesoresAsistenciasPDFAction(), GenerarUltimaAsistenciasProfesoresAIFrentePDFAction(), GenerarUltimaAsistenciasProfesoresPDFAction(), GenerarPlanificacionPDFAction() para exportar en formato PDF los reportes de asistencia (estudiantes y profesores) y la planificación de la guardia.	

Tabla 44: Tarea de ingeniería 9

Tarea de Ingeniería	
<b>No. De la tarea:</b> 9	<b>No. De la HU:</b> 6
<b>Nombre de la tarea:</b> Mostrar planificación.	

<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 1/04/2015	<b>Fecha fin:</b> 3/04/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Con el empleo del código php app/console doctrine:generate:crud, generar el crud necesario para mostrar la planificación total del servicio de guardia.	

**Tabla 45: Tarea de ingeniería 10**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 10	<b>No. De la HU:</b> 7
<b>Nombre de la tarea:</b> Mostrar mi planificación.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 6/04/2015	<b>Fecha fin:</b> 8/04/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Implementar el método miPlanAction() que se permitirá mostrar la planificación del usuario autenticado en el sistema.	

**Tabla 46: Tarea de ingeniería 11**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 11	<b>No. De la HU:</b> 8
<b>Nombre de la tarea:</b> Mostrar reporte.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 9/04/2015	<b>Fecha fin:</b> 13/04/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Implementar el método mostrarReporteAction() para publicar la información sobre la asistencia de los usuarios del servicio de guardia.	

**Tabla 47: Tarea de ingeniería 12**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 12	<b>No. De la HU:</b> 9
<b>Nombre de la tarea:</b> Control de asistencia.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 14/04/2015	<b>Fecha fin:</b> 16/04/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Con el empleo del código php app/console doctrine:generate:crud, generar el crud necesario para actualizar la asistencia de los usuarios.	

Tabla 48: Tarea de ingeniería 13

Tarea de Ingeniería	
<b>No. De la tarea:</b> 13	<b>No. De la HU:</b> 10
<b>Nombre de la tarea:</b> Buscar integrantes para solicitar intercambio.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 17/04/2015	<b>Fecha fin:</b> 21/04/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Implementar el método buscarEstudianteAction().	

Tabla 49: Tarea de ingeniería 14

Tarea de Ingeniería	
<b>No. De la tarea:</b> 14	<b>No. De la HU:</b> 11
<b>Nombre de la tarea:</b> Intercambiar turno.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 22/04/2015	<b>Fecha fin:</b> 24/04/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Implementar el método intercambiarTurnoAction() que va a permitir enviar	

solicitudes de cambio de turno entre los usuarios.

**Tabla 50: Tarea de ingeniería 15**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 15	<b>No. De la HU:</b> 12
<b>Nombre de la tarea:</b> Listar intercambios.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 27/04/2015	<b>Fecha fin:</b> 29/04/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Implementar el método listarIntercambioAction() para mostrar los intercambios disponibles para un usuarios.	

**Tabla 51: Tarea de ingeniería 16**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 16	<b>No. De la HU:</b> 13
<b>Nombre de la tarea:</b> Insertar usuario y rol.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 29/04/2015	<b>Fecha fin:</b> 1/05/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Generar un crud a la clase FosUser para asignar roles a los usuarios locales y generar un crud a la clase webserviceusuarios para asignar roles a los usuarios del dominio uci e insertarlos en el sistema.	

**Tabla 52: Tarea de ingeniería 17**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 17	<b>No. De la HU:</b> 13
<b>Nombre de la tarea:</b> Mostrar usuarios con roles asignados.	

<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 29/04/2015	<b>Fecha fin:</b> 1/05/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Generar un crud a la clase FosUser para mostrar los usuarios que poseen roles en el sistema.	

**Tabla 53: Tarea de ingeniería 18**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 18	<b>No. De la HU:</b> 13
<b>Nombre de la tarea:</b> Eliminar usuarios con rol asignado.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 29/04/2015	<b>Fecha fin:</b> 1/05/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Generar un crud a la clase FosUser para eliminar los usuarios que poseen roles en el sistema.	

**Tabla 54: Tarea de ingeniería 19**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 19	<b>No. De la HU:</b> 13
<b>Nombre de la tarea:</b> Modificar el rol de un usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3
<b>Fecha inicio:</b> 29/04/2015	<b>Fecha fin:</b> 1/05/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Generar un crud a la clase FosUser para modificar los roles de los usuarios en el sistema.	

**Tabla 55: Tarea de ingeniería 20**

Tarea de Ingeniería	
<b>No. De la tarea:</b> 20	<b>No. De la HU:</b> 14
<b>Nombre de la tarea:</b> Importar usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 7
<b>Fecha inicio:</b> 1/05/2015	<b>Fecha fin:</b> 8/05/2015
<b>Programador responsable:</b> Sifredo Escalona Ramos y Wilson Ojeda Grass	
<b>Descripción:</b> Implementar el método importarExcelAction() para añadir a la base de datos del sistema la información de los usuarios contenida en el archivo Excel.	

**Anexo 4: Pruebas de aceptación**

**Tabla 56: Prueba de aceptación No. 4**

Pruebas de aceptación	
<b>Número:</b> 4	<b>No. HU:</b> 2
<b>Nombre:</b> Mostrar usuarios.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario visualizar todos los usuarios del sistema.	
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol administrador y deben existir usuarios en la base de datos del sistema.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Gestionar usuarios" en el menú principal. Luego la aplicación muestra la lista de todos los usuarios en el sistema.	
<b>Resultado esperado:</b> El sistema permite visualizar todos los usuarios del SGOE.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 57: Prueba de aceptación No. 5**

Pruebas de aceptación	
<b>Número:</b> 5	<b>No. HU:</b> 3

<b>Nombre:</b> Crear brigada.
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario crear las brigadas del SGOE.
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Administrador y deben existir usuarios en la base de datos del sistema.
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Crear Brigadas Estudiantes" en el menú "Gestionar Guardia" y especificara la cantidad de brigadas. Finalmente podrá confirmar esta acción en la opción "Crear Brigadas".
<b>Resultado esperado:</b> El sistema crea las brigadas del SGOE.
<b>Evaluación de la prueba:</b> Satisfactoria.

**Tabla 58: Prueba de aceptación No. 6**

Pruebas de aceptación	
<b>Número:</b> 6	<b>No. HU:</b> 4
<b>Nombre:</b> Generar planificación de estudiantes.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario realizar la planificación del servicio de guardia para los estudiantes.	
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Administrador y deben existir usuarios en la base de datos del sistema.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Planificar Guardia Estudiantes ", en el menú "Gestionar Guardia" y después puede especificar las zonas, turnos, fecha de inicio y la fecha con la cantidad de días que no estará disponible el servicio de guardia.	
<b>Resultado esperado:</b> El sistema realiza la planificación del SGOE.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 59: Prueba de aceptación No. 7**

Pruebas de aceptación	
<b>Número:</b> 7	<b>No. HU:</b> 4
<b>Nombre:</b> Generar planificación de profesores.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario realizar la planificación del	

servicio de guardia para los profesores.
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Administrador y deben existir usuarios en la base de datos del sistema.
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Planificar Guardia Profesores", en el menú "Gestionar Guardia" y después puede especificar las zonas, fecha de inicio y la fecha con la cantidad de días que no estará disponible el servicio de guardia.
<b>Resultado esperado:</b> El sistema realiza la planificación del SGOE.
<b>Evaluación de la prueba:</b> Satisfactoria.

**Tabla 60: Prueba de aceptación No. 8**

Pruebas de aceptación	
<b>Número:</b> 8	<b>No. HU:</b> 5
<b>Nombre:</b> Generar PDF.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario exportar la información en formato PDF.	
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Administrador.	
<b>Pasos de ejecución:</b> El usuario puede seleccionar las opciones "Planes de Guardia" en el menú "Gestionar Guardia" o "Reportes de Asistencia" en el menú "Control y Reportes" y el sistema le brindará al usuario la opción de descargar la información en formato PDF.	
<b>Resultado esperado:</b> El sistema exporta la información en formato PDF.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 61: Prueba de aceptación No. 9**

Pruebas de aceptación	
<b>Número:</b> 9	<b>No. HU:</b> 6
<b>Nombre:</b> Mostrar planificación.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario visualizar la planificación del SGOE.	
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Administrador y se debe	

haber realizado la planificación.
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Planes de Guardia" en el menú "Gestionar Guardia" y el sistema le permitirá visualizar la planificación.
<b>Resultado esperado:</b> El sistema muestra la planificación del SGOE.
<b>Evaluación de la prueba:</b> Satisfactoria.

**Tabla 62: Prueba de aceptación No. 10**

Pruebas de aceptación	
<b>Número:</b> 10	<b>No. HU:</b> 7
<b>Nombre:</b> Mostrar mi planificación.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario ver su planificación.	
<b>Condiciones de ejecución:</b> La planificación debe haberse realizado.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Mi Planificación" en el menú "Opciones de usuario" y el sistema le permitirá visualizar su planificación del SGOE.	
<b>Resultado esperado:</b> El sistema muestra la planificación del usuario autenticado.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 63: Prueba de aceptación No. 11**

Pruebas de aceptación	
<b>Número:</b> 11	<b>No. HU:</b> 8
<b>Nombre:</b> Mostrar reporte.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario ver el reporte de asistencia del SGOE.	
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Administrador.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Reportes de Asistencia", en el menú "Control y Reportes", y puede visualizar en la aplicación el reporte de asistencia.	
<b>Resultado esperado:</b> El sistema muestra la asistencia de los usuarios.	

**Evaluación de la prueba:** Satisfactoria.

**Tabla 64: Prueba de aceptación No. 12**

Pruebas de aceptación	
<b>Número:</b> 12	<b>No. HU:</b> 9
<b>Nombre:</b> Control de asistencia.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario controlar la asistencia al servicio de guardia.	
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Administrador.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Reportes de Asistencia" en el menú principal. Luego puede acceder a las categorías "Estudiante", "ProfesorAlFrente" o "Profesor" y tendrá acceso a la opción "Modificar" en la cual cambia el valor de la asistencia del usuario.	
<b>Resultado esperado:</b> El sistema permite cambiar el valor de la asistencia.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 65: Prueba de aceptación No. 13**

Pruebas de aceptación	
<b>Número:</b> 13	<b>No. HU:</b> 10
<b>Nombre:</b> Buscar integrante del servicio de guardia.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario buscar otro integrante del servicio de guardia.	
<b>Condiciones de ejecución:</b> El usuario a buscar debe existir en la base de datos.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Seleccionar Usuario" en el menú "Opciones de Usuario" y puede buscar a otro integrante del servicio de guardia.	
<b>Resultado esperado:</b> El sistema muestra el usuario buscado.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 66: Prueba de aceptación No. 14**

Pruebas de aceptación
-----------------------

<b>Número:</b> 14	<b>No. HU:</b> 11
<b>Nombre:</b> Intercambiar turno.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario intercambiar el turno de guardia.	
<b>Condiciones de ejecución:</b> El usuario con el que se va a intercambiar debe estar disponible.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Solicitar Intercambio" en el menú "Opciones de Usuario" y después ingresar el usuario con el que desea intercambiar y enviar la notificación.	
<b>Resultado esperado:</b> El sistema envía el intercambio de turno.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 67: Prueba de aceptación No. 15**

Pruebas de aceptación	
<b>Número:</b> 15	<b>No. HU:</b> 12
<b>Nombre:</b> Listar intercambios.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario listar los intercambios disponibles.	
<b>Condiciones de ejecución:</b> El usuario a intercambiar debe existir en la base de datos y ser de la misma categoría.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Listar Intercambio" en el menú principal. Luego podrá visualizar las peticiones de intercambios disponibles en el sistema y aceptarlas o denegarlas.	
<b>Resultado esperado:</b> El sistema lista los intercambios realizados.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 68: Prueba de aceptación No. 16**

Pruebas de aceptación	
<b>Número:</b> 16	<b>No. HU:</b> 13
<b>Nombre:</b> Insertar rol a un usuario.	

<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario insertar un usuario con un rol.
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Super-administrador.
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Asignar Roles" en el menú principal. Luego accede a la opción "Crear" en la que tendrá acceso a añadir un usuario con rol.
<b>Resultado esperado:</b> El sistema inserta un usuario con un rol.
<b>Evaluación de la prueba:</b> Satisfactoria.

**Tabla 69: Prueba de aceptación No. 17**

Pruebas de aceptación	
<b>Número:</b> 17	<b>No. HU:</b> 13
<b>Nombre:</b> Mostrar usuarios y su rol.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario ver los usuarios a los que se les asigno un rol en el sistema.	
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Super-administrador.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Asignar Roles" en el menú principal. Luego accede a la opción "Mostrar" y el sistema mostrará el usuario y su rol en el sistema.	
<b>Resultado esperado:</b> El sistema muestra el usuario y su rol.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 70: Prueba de aceptación No. 18**

Pruebas de aceptación	
<b>Número:</b> 18	<b>No. HU:</b> 13
<b>Nombre:</b> Eliminar usuario con rol.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario eliminar los usuarios a los que se les asigno un rol en el sistema.	
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Super-administrador.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Asignar roles" en el menú principal. Luego accede a la opción "Modificar" y selecciona la opción "Eliminar" en la que el sistema le	

permitirá eliminar un usuario con rol.
<b>Resultado esperado:</b> El sistema elimina un usuario con rol.
<b>Evaluación de la prueba:</b> Satisfactoria.

**Tabla 71: Prueba de aceptación No. 19**

Pruebas de aceptación	
<b>Número:</b> 19	<b>No. HU:</b> 13
<b>Nombre:</b> Modificar rol.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario modificar el rol de un usuario.	
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Super-administrador.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Asignar Roles" en el menú principal. Luego accede a la opción "Modificar" y el sistema le permitirá cambiar el rol de un usuario.	
<b>Resultado esperado:</b> El sistema modifica el rol de un usuario.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 72: Prueba de aceptación No. 20**

Pruebas de aceptación	
<b>Número:</b> 20	<b>No. HU:</b> 14
<b>Nombre:</b> Importar usuario.	
<b>Descripción:</b> Prueba para la funcionalidad que permite al usuario importar la información del personal del servicio de guardia a la base de datos del sistema.	
<b>Condiciones de ejecución:</b> El usuario autenticado debe tener rol Super-administrador.	
<b>Pasos de ejecución:</b> El usuario selecciona la opción "Importar usuarios" en el menú principal. Luego el sistema le permite añadir el archivo con la información de los usuarios y cargar el fichero en la base de datos.	
<b>Resultado esperado:</b> El sistema añade la información de los usuarios desde el fichero Excel.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

