

Universidad de las Ciencias Informáticas

Facultad 3



**Aplicación web para la Gestión de los
Laboratorios del Centro de Gobierno Electrónico**

**Trabajo de Diploma para optar por el título de
Ingeniero Informático**

Autor: Mailenys Rodríguez Paez.

Tutor: Ing. Yosvany Márquez Ruiz.

Co-tutor: Ing. Yamilé Stivens Portela.

La Habana, Cuba

Junio 2013

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas hacer uso del mismo en su beneficio.

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

Mailenys Rodríguez Paez

Ing. Yosvany Márquez Ruiz.

Ing. Yamilé Stivens Portela.

AGRADECIMIENTOS

A mi papín, por transmitirme siempre seguridad y por brindarme su apoyo e incondicional.

A mi mamita, por depositar en mí su confianza y por apoyarme en todas mis decisiones.

A mi tía Mimi y mi abuelita María por ser la más linda del mundo y por siempre defenderme por encima de cualquier cosa.

A mi novio y amigo por aguantarme todos estos meses que han sido muy importantes en mi vida.

A mis tías Maricel (Pipa) y Marilyn (Millo) por estar ahí cuando necesité su apoyo y comprensión.

A mis primas y primos Mairelys, Patricia, Amandita y Danielito por siempre brindarme esa alegría que me permite seguir luchando por lo que quiero.

A mi gordita la más linda punto com que aunque está muy lejos sé que nunca me olvida. Te quiero mucho Gemita.

A mi tata Arelys que me ha ayudado mucho y siempre ha estado presente en las buenas y en las malas.

A todos mis compañeros de aula que me han aguantado estos cinco años.

A mis tutores Yosvany y Yamile que siempre me ayudaron en todo los momentos que los molestaba.

A todas mis amistades del edificio 16.gracias por brindarme ese apoyo incondicional.

A Yordanys que fue el mejor jefe de proyecto que he tenido en mi vida y mi corrector ortográfico te quiero.

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL.

RESUMEN

En el Centro de Gobierno Electrónico se necesita elevar el nivel de seguridad de sus activos, localizados en los diferentes laboratorios de producción. En tal sentido se desarrollan un conjunto de actividades como son el control de acceso de las personas, la gestión de los activos y la gestión de incidencias. Estas tareas se complementan con las actividades propias de la gestión de los laboratorios y tributan en gran medida con la seguridad deseada en los laboratorios del centro. Todas ellas en su conjunto favorecen a la gestión de la seguridad informática en dichas áreas. Es por ello que se realizó un profundo y detallado estudio del arte donde se analizaron sistemas existentes tanto en Cuba como en el resto del mundo que incluyen dentro de sus funcionalidades las actividades que son hoy objetivo fundamental para el centro. Esto permitió entender y llegar a conclusiones claras del sistema que se desea desarrollar. Para ello se seleccionaron las herramientas, tecnologías y lenguajes de programación utilizados para realizar la construcción del sistema. Se identificaron los requisitos funcionales y no funcionales, para así trazar la línea base de la arquitectura, realizar el diseño de la base de datos e implementar el sistema a partir de los requisitos obtenidos, y con ello realizar las validaciones pertinentes a la solución propuesta.

Palabras claves.

Seguridad, Control de acceso, Gestión de Activos, Gestión de Incidencia.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	7
1.1- Introducción	7
1.2- Conceptos fundamentales	7
1.3- Norma Cubana (NC) ISO/IEC 17799: 2007	8
1.4- Metodología de desarrollo de software	16
1.5- Notación para el modelado del negocio	18
1.6- Marco de trabajo.....	19
1.7- Lenguajes utilizados	22
1.8- Herramientas seleccionadas para la solución	24
1.9- Conclusiones parciales	27
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN	28
2.1- Introducción	28
2.2- Modelado de negocio.....	28
2.3- Requisitos.....	31
2.4 -Análisis	39
2.5- Diseño	43
2.6- Implementación.....	56
2.7- Conclusiones parciales	59
CAPÍTULO 3: VALIDACIÓN	60
3.1- Introducción	60
3.2- Validación del modelado de negocio.....	60

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

3.3- Validación de requisitos	60
3.4- Validaciones del diseño	62
3.5- Pruebas de software	65
3.6- Análisis cuantitativo de la solución	66
3.7- Conclusiones parciales	70
CONCLUSIONES	71
RECOMENDACIONES	72
REFERENCIAS BIBLIOGRÁFICAS	73

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

INTRODUCCIÓN

En el mundo de hoy, es cada vez más evidente que las Tecnologías de la Información y las Comunicaciones (TIC) se encuentran presentes en todas las esferas de la sociedad. El país no se encuentra ajeno a este desarrollo, enfrentando el reto de informatizar la sociedad con vistas a integrarse a la infraestructura global de la información haciendo uso de la informática y las comunicaciones.

Por esta razón, el estado cubano se encuentra inmerso en lo que se ha llamado la “Informatización de la Sociedad Cubana”, mediante este proyecto se aplican las TIC a las diferentes esferas y sectores de la sociedad, para lograr como resultado una mayor eficiencia y eficacia con la optimización de los recursos y el logro de una mayor productividad y competitividad.

La Universidad de las Ciencias Informáticas (UCI), surge en medio de este deseo de crecer en el desarrollo tecnológico, contribuyendo a la formación de profesionales capaces de explotar las nuevas tecnologías, implementando ideas y proyectos que eleven el uso de esta rama a nivel empresarial, social y gubernamental.

Para darle cumplimiento a estas ideas surge dentro de la UCI el Centro de Gobierno Electrónico (CEGEL), con la misión de satisfacer necesidades de clientes gubernamentales mediante el desarrollo de productos, servicios y soluciones integrales de alta confiabilidad, calidad, competitividad, fidelidad y eficiencia. El centro aspira a ser de referencia nacional en el desarrollo de proyectos, servicios y soluciones informáticas para la gestión de las áreas de gobierno, con una elevada integración en los procesos de producción, investigación y formación de pregrado y postgrado, respaldado por un personal altamente calificado (1).

Como toda entidad vinculada a las Tecnologías de la Información; el centro CEGEL se enfrenta a amenazas de seguridad procedentes de una amplia variedad de fuentes, incluyendo fraudes informáticos, espionaje, sabotaje, vandalismo, incendios e inundaciones. Hoy en día estas amenazas figuran como elementos que atentan contra el cumplimiento de los objetivos estratégicos del centro, por lo tanto es importante garantizar la gestión de la seguridad informática en el área. (2).

En la actualidad la mayor concentración de activos que tiene el centro están ubicados en los laboratorios de producción, a los cuales acceden un número considerable de personas, dígame trabajadores, estudiantes del centro, además de visitantes externos relacionados directamente o no con los proyectos productivos que se desarrollan, dentro de los trabajadores se incluyen un grupo de técnicos generales que se encargan

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

de conservar los activos en conjunto con un grupo determinado de profesores y especialistas. Entre las tareas que realizan estas personas se encuentran:

- Hacer cumplir las normas de comportamiento y uso de los medios de cada uno de los laboratorios y las políticas de seguridad informática definidas en la UCI y el propio centro.
- Llevar a cabo una correcta gestión de los activos.
- Llevar un estricto control del movimiento de cada uno de los activos tecnológicos y actualización de los mismos en el inventario de cada local.
- Realizar una correcta recepción y entrega de la responsabilidad sobre los medios que se encuentran bajo su custodia por cada local.
- Establecer en cada momento la responsabilidad sobre los activos.
- Controlar el acceso físico de personas a los locales que custodian.
- Gestión de las incidencias de cualquier índole que se sucedan durante la jornada laboral y notificar de acuerdo al tipo de incidencia al personal pertinente para atender cada situación.

A pesar de que existe claridad en cómo desempeñar cada una de estas tareas por parte del personal, se han detectado en reiteradas ocasiones diversas irregularidades que atentan contra la integridad de la gestión de la seguridad, principalmente porque estas actividades propias del proceso de gestión de laboratorios de producción no logran realizarse de manera eficiente en todos los casos y así se evidencia durante los diagnósticos realizados, donde la dirección del centro ha detectado varias deficiencias como son:

- La relación de activos propios del laboratorio, en la mayoría de las ocasiones se encuentran desactualizados porque estos se encuentran en un listado impreso que se ubica detrás de la puerta de cada local, y por tanto ante cualquier traslado sea temporal o permanente dicho listado queda desactualizado.
- No existe un registro de los movimientos temporales o permanentes de los activos y por tanto ante un seguimiento al mismo no se conoce la ubicación exacta, salvo que se revisen todos los locales del centro en caso de que el traslado sea interno.
- La identificación del personal que tienen acceso al laboratorio se realiza de manera irregular debido a la lentitud del proceso. El cual se realiza de forma manual y la dinámica con que suceden dichos accesos hace que la actividad sea monótona lo que provoca que en reiteradas ocasiones el control

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

de acceso se realice a través de un reconocimiento facial de la persona por parte del técnico dejando de registrarse el acceso.

- Se dificulta conocer con exactitud las personas que en cierto momento han accedido a los laboratorios a pesar de que existe un registro de acceso en cada local.
- Los listados de acceso a los laboratorios sufren modificaciones, en las que se incorporan o eliminan los nombres de estudiantes y profesores que acceden a estos locales y cuando esto sucede no son notificados al personal que atiende esta actividad.
- La lentitud del proceso de tramitación de las incidencias provoca que pierda objetividad informarlas a las personas involucradas pasado un tiempo determinado, debido a que puede darse el caso que dichas incidencias no se detecten el mismo día y por tanto se pierde el momento exacto en que ocurrió.
- En muchas ocasiones ocurren incidencias que debían ser notificadas por estudiantes y profesores al personal técnico para que se registren y notifiquen, sin embargo no se hace por varias razones, dentro de ellas por el desconocimiento de que es una incidencia y que no, porque no se ha creado una cultura institucional acerca de la seguridad informática y por la irresponsabilidad de algunos pocos.
- Al no registrarse todas las incidencias se pierde información valiosa para la identificación, evaluación y tratamiento de riesgos propios de la gestión de seguridad informática.
- Durante la actividad de entrega y recepción de los laboratorios se pierde información importante relacionada con el estado de los medios y mecanismos de seguridad, ejemplo de ello el estado de los sellos que son ubicados en las estaciones de trabajo y servidores con el objetivo que se pueda detectar a través de chequeos sistemáticos el acceso a sus componentes como memorias, disco duro, entre otros.

Sumado a todas estas deficiencias, se tienen otras relacionadas con el gasto de recursos de materiales de oficina como son papel, tinta de impresión, lapiceros, entre otros, principalmente para la confección y actualización de los documentos que forman parte de estas actividades, como por ejemplo el registro de entrega y recepción de los locales y el registro de incidencias. Por todo lo antes expuesto se hizo necesario plantearse el siguiente **problema a resolver**: ¿Cómo contribuir con las actividades de control de acceso y de gestión de los activos en los laboratorios de producción de CEGEL para mejorar la gestión de la seguridad informática?

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

Por tanto el **objeto de estudio** de este trabajo es el proceso de gestión de seguridad informática en los laboratorios de producción encaminado a cumplir el **objetivo General**: Desarrollar una aplicación web para informatizar las actividades de control de acceso y gestión de los activos en los laboratorios de CEGEL para contribuir a la mejora de la gestión de la seguridad informática. El **campo de acción** queda enmarcado en las actividades de control de acceso y gestión de la información de activos en los Laboratorios de CEGEL. En correspondencia con el problema y con la intención de darle cumplimiento al objetivo propuesto, se **defiende la idea** de que si se desarrolla una aplicación web para informatizar las actividades de control de acceso y gestión de los activos en los laboratorios de producción de CEGEL se podrá mejorar la gestión de la seguridad informática en estas áreas.

Del objetivo general de la investigación se derivan los siguientes **Objetivos Específicos**:

1. Elaborar el marco teórico de la investigación para fundamentarla con base al planteamiento del problema que se ha realizado.
2. Desarrollar la propuesta de la solución que permitirá dar respuesta a la problemática planteada.
3. Validar la solución propuesta para evaluar el cumplimiento de los objetivos planteados en la investigación.

Para darle cumplimiento al objetivo trazado se han planteado las siguientes **Tareas a cumplir**:

1. Elaboración del marco conceptual de la investigación.
2. Estudio de sistemas existentes relacionados con el objeto de estudio.
3. Estudio del proceso de gestión de los laboratorios del Centro de Gobierno Electrónico (CEGEL).
4. Selección de la metodología de desarrollo de software a emplear.
5. Definición de las herramientas y tecnologías a utilizar para el desarrollo de la propuesta de solución.
6. Obtención y validación del modelo de negocio.
7. Levantamiento y validación de los requisitos del sistema.
8. Obtención del modelo de análisis y diseño del sistema.
9. Validación del diseño del sistema.
10. Implementación de la solución.

11. Validación del software.

Posibles resultados

- Se desarrollará una aplicación web para el control de acceso y la gestión de los activos de los Laboratorios del Centro de Gobierno Electrónico.

Entre los métodos de investigación científica existentes fueron utilizados los **Métodos teóricos**:

Análítico –Sintético: para realizar una revisión de sitios web, documentos y tesis, todos relacionados con el tema, de los cuales se obtuvo la información específica que se necesitaba para el desarrollo de la investigación.

Histórico –Lógico: para realizar un estudio de las tecnologías y herramienta desde un entorno más progresivo viendo así sus antecedentes históricos y tendencias actuales con el fin se seleccionar las más apropiadas para darle cumplimiento al objetivo general de la presente investigación.

Modelación: Se utiliza con el objetivo de reproducir la interacción de los objetos en la vida real mediante la creación de prototipos funcionales y la representación de los requisitos del producto en los artefactos necesarios para el desarrollo del producto.

Los **Métodos empírico** estudiados fueron:

Entrevista: Se utiliza con el objetivo identificar las necesidades concretas que tiene el cliente y poderlas traducir en los requisitos del producto.

Observación: a través de este método se pudo conocer la particularidad de la problemática definida, lo que ayudó al planteamiento del problema a resolver, además de permitir conocer el proceso definido como objeto de estudio, lo cual influye a la hora de tener un conocimiento más detallado de lo que se quiere, lo que hace falta hacer y cómo hay que hacerlo.

Medición: se utiliza para medir la calidad de la especificación de los requisitos y el grado de ambigüedad de estos, además de obtener una medida de la calidad del diseño para su validación.

El presente trabajo de diploma posee la siguiente estructura:

Capítulo 1: Fundamentación Teórica

Este capítulo contiene una base teórica que permite entender el problema a solucionar, recoge los conceptos fundamentales que se deben tener claros para la correcta comprensión del trabajo, también se definen la

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

metodología, lenguajes y herramientas que servirán de apoyo para la realización de la presente investigación. Se expone el marco conceptual sobre el que gira la propuesta de solución.

Capítulo 2: Propuesta de solución

Este capítulo recoge lo relacionado con la propuesta de solución de la presente investigación. Estará compuesto por los artefactos que se generan según la metodología seleccionada como son: el diagrama de caso de uso del sistema, el modelo de datos, los diagramas de clases del diseño, la arquitectura desarrollada y los patrones arquitectónicos utilizados para darle solución a la problemática planteada. Se especifican los estándares de codificación, el diagrama de despliegue, el diagrama de componentes y la descripción de las principales funcionalidades a informatizar.

Capítulo 3: Validación y prueba

En este capítulo se efectúan las validaciones de la solución propuesta. Las mismas se realizarán mediante pruebas de caja negra y haciendo uso de métricas, con las cuales se evaluará la solución haciendo un resumen de los principales resultados obtenidos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1- Introducción

En el presente capítulo se brinda una visión general de los aspectos relacionados con los Sistemas de Control de Acceso, la Gestión de Activos Tecnológicos y los Registros de Incidencia según está estipulado en la Norma Cubana ISO/IEC 17799:2007. Se describen los principales conceptos asociados al problema que son necesarios para entender el negocio y la propuesta de solución. Serán objetos de comparación las tecnologías punteras del mercado en este sector, su incidencia y relación con otros sistemas. Se describen los procesos del negocio que se relacionan con el objeto de estudio de este trabajo. A lo largo del presente trabajo se seleccionan varias herramientas y tecnologías, por lo que se efectúa una descripción de las mismas con el objetivo de conocerlas y brindar una panorámica de la solución.

1.2- Conceptos fundamentales

Activo: es un bien tangible o intangible que posee una empresa o persona natural, es un recurso o bien económico propiedad de una empresa, con el cual se obtienen beneficios (3).

Es todo aquello que tenga valor para la organización (2).

Seguridad: es una característica de cualquier sistema informático o no que no se registren peligros, daños, ni riesgos. Una cosa segura es algo firme, cierto e indubitable (4).

Protección: es la acción y efecto de proteger (resguardar, defender o amparar a algo o alguien). La protección es un cuidado preventivo ante un eventual riesgo o problema (4).

Control: son todos los mecanismos, acciones, herramientas que se realizan para detectar la presencia de errores y la realización de inspecciones o pruebas de muestreo para verificar que las características de un elemento determinado sean óptimas (4).

Acceso: es un término que puede utilizarse en todos los ámbitos de los servicios de interés público. Se refiere a una meta a conseguir (4).

El **Control de Acceso** es el proceso de gestionar la entrada y salida de personas o medios a un lugar determinado, registrándose datos necesarios para garantizar el cumplimiento de una política de permisos que consolide la seguridad del lugar. El control de acceso es necesario porque es la primera barrera de defensa contra personas no deseadas, que pudieran causar algún tipo de problema dentro de la institución (4).

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

Sistema: es un todo integrado, compuesto de estructuras diversas, interactuantes y especializadas. La complejidad de la combinación está implícita, por lo que es una colección organizada para cumplir un objetivo específico (4).

Identificación: es la acción por parte de un usuario de presentar su identidad a un sistema, para ello se usa un identificador de usuario. Establece que el usuario es responsable de las acciones que lleve a cabo en el sistema, lo que se relaciona con los registros de auditorías que permiten guardar las acciones realizadas dentro del sistema y rastrearlas hasta el usuario autenticado (4).

Información: es un activo que es esencial al negocio de una organización y requiere en consecuencia una protección adecuada (5).

Protección de la información: cuando se habla de protección de la información se habla de proteger el funcionamiento adecuado de la empresa, sobre todo en casos de fallos informáticos, asimismo, se pueden evitar pérdidas financieras provocadas por la desaparición de archivos, bases de datos, balances financieros (5).

Seguridad de la información: es el conjunto de medidas preventivas y reactivas de las organizaciones y de los sistemas tecnológicos que permitan resguardar y proteger la información buscando mantener la confidencialidad, la disponibilidad e integridad de la misma (6).

Autenticación: es la verificación de que el usuario que trata de identificarse es válido, se implementa con una contraseña en el momento de iniciar una sesión (7).

Autorización: es la concesión permiso o derecho dado al usuario para hacer algo en la aplicación (7).

Un **proceso** es un conjunto de actividades o eventos (coordinados u organizados) que se realizan o suceden (alternativa o simultáneamente) bajo ciertas circunstancias con un fin determinado (8).

1.3- Norma Cubana (NC) ISO/IEC 17799: 2007

La Norma Cubana establece recomendaciones y principios generales para iniciar, implantar, mantener y mejorar la gestión de la seguridad informática en una organización. Los objetivos señalados en esta norma internacional proporcionan recomendaciones generales sobre las metas comunes aceptadas para la gestión de la seguridad informática.

Los objetivos de control y los controles de la norma internacional están pensados para ser implementados a fin de alcanzar los requisitos identificados por una evaluación del riesgo. Puede servir como

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

recomendación práctica para desarrollar normas de seguridad de la organización, una práctica efectiva de la gestión de la misma y permitir ayudar a construir confianza en las actividades entre organizaciones.

Esta norma contiene 11 cláusulas de control de la seguridad que en su conjunto contienen un total de 39 categorías principales de seguridad y una cláusula introductoria a la evaluación y tratamiento de riesgos.

Estas cláusulas son (2):

- Política de Seguridad.
- Organización de la Seguridad de la Información.
- Gestión de Activos.
- Seguridad de Recursos Humanos.
- Seguridad Física y del Ambiente.
- Gestión de Comunicaciones y Operaciones.
- Control de Acceso.
- Adquisición, Desarrollo y Mantenimiento de los Sistemas de Información.
- Gestión de incidentes de la seguridad de la información.
- Gestión de la continuidad del negocio.

Muchas de estas cláusulas se van más allá del alcance de la investigación. Son objetivo de estudio la Gestión de activos, la Seguridad física y del ambiente, y la Gestión de incidentes de la seguridad de la información. A continuación se procede a realizar una breve descripción de cada una de ellas según las plantea la Norma Cubana.

La **Gestión de activos** debe regirse por: implementar y mantener una adecuada protección sobre los activos de la organización, que todos los activos han de ser identificados y deben realizar y mantener el inventario de los más importantes.

La **Seguridad Física y del Ambiente** según la NC plantea que el objetivo principal es evitar accesos físicos no autorizados, evitar daños e interferencias en las instalaciones y proteger la información de la organización. Propone también utilizar perímetros de seguridad tales como barreras, paredes, puertas de entrada controladas por tarjetas o recepcionistas, o controles de acceso apropiados para la institución.

La **Gestión de incidentes de la seguridad de la información** según plantea la NC debe asegurar que las debilidades y eventos de seguridad de la información asociados a sistemas son comunicados de

manera de poder tomar acciones correctivas a tiempo, que se establezca un procedimiento formal de reportes donde todos los involucrados en la incidencia estén al tanto de ellas.

Todo lo anterior expuesto será la integración que se espera llevar a cabo con la realización del Sistema de Gestión de los Laboratorios del Centro de Gobierno Electrónico. A continuación se procede a realizar el estudio de sistemas relacionados con el campo de acción y con cada uno de las cláusulas pautadas por la Norma Cubana.

1.3.1- Gestión de activos

La Gestión de activos es una disciplina que busca gestionar todo lo relacionado con los activos físicos de una organización con el fin de tener su control (2). Entre los sistemas que se dedican a dichas actividades se encuentran los que se detallan a continuación.

1.3.1.1- Sistema de Información de Activos SIGA

Programa de Gestión de Activos (PROGA): El objetivo de este programa es la maximización económica y social de los bienes inmuebles del Estado. El PROGA busca implantar prácticas eficientes de gestión inmobiliaria en las entidades del nivel central del gobierno, con el fin de disminuir la tenencia de activos improductivos en poder de entidades y promover un uso racional. Es por ello que PROGA crea el Sistema de Información de Activos SIGA (9).

El SIGA es un sistema que posee las siguientes funcionalidades (9):

- Centralizar la información.
- Contar con la información depurada y actualizada.
- Lograr una alta disponibilidad de la información.
- Agilizar la toma de decisiones.
- Facilitar las consultas.
- Eliminar los ciclos de producción de la información.
- Evitar la redundancia de esfuerzos.
- Disminuir los costos de la administración de la información.
- Brindar herramientas a los usuarios, que faciliten los análisis que requiere el PROGA.

1.3.1.2- (IBM Máximo) IBM (International Business Machines)

IBM Máximo es un sistema basado en una sola plataforma de software, que ofrece una visión global de todos los tipos de activos de la empresa. Esta perspectiva integral le permite ver los activos, identificar todo su potencial sin explotar, obtener los conocimientos y el control necesarios para alinear con precisión los objetivos de la organización con los objetivos globales de la empresa. Compuesto de seis módulos de gestión clave (activos, trabajo, servicio, contrato, materiales y gestión de aprovisionamiento). Con su puesta en marcha permite el aumento del rendimiento de los activos, la disminución de los costes y el riesgo, el aumento de la productividad, la mejora en la toma de decisiones relacionada con activos, el aumento de la capacidad de respuesta de la prestación de servicios de activos y de los ingresos, una simplificación de los esfuerzos de conformidad normativa y la disminución de coste total de propiedad (10).

1.3.1.3- GLPI (Libre Parque Informático de Gestión)

Muchas veces según van creciendo las empresas se necesitan herramientas para organizar toda la gestión del inventario de cada equipo. Algo que ayude a tener controlado todo lo que está instalado en cada local, tanto a nivel de software como a nivel de hardware. GLPI es una completa aplicación de código abierto para gestionar el inventario que puede ayudar en esta tarea. La aplicación está basada en PHP¹, MySQL² y Apache³, lo que permite una administración vía navegador web que ayuda con la gestión diaria de la base de datos de inventario, facilita también la tarea de acceso desde cualquier equipo de la empresa sin necesidad de tener que estar físicamente delante del equipo que tiene instalada la aplicación. Esto será muy útil para mantener el inventario al día, puesto que mientras esté trabajando en una máquina, se puede comprobar que el estado que tiene en el inventario corresponde con lo que está instalado en el equipo (11).

Entre las principales características de GPLI se pueden destacar (11):

- **Inventario de los equipos**, con posibilidad de incluir componentes de hardware como discos duros, y otros componentes de hardware.

1 Hypertext Preprocessor: Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

2 MySQL: Sistema de gestión de bases de datos relacional.

3 Apache: servidor web de software libre.

- **Inventario de periféricos** asociados a los equipos, ya sean monitores, ratones, teclados, altavoces, impresoras, entre otros.
- **Inventario de red** donde se pueden establecer todos los criterios de conexión de los equipos, IP, dirección MAC de la tarjeta de red, VLAN⁴ configuradas.

1.3.1.4- Análisis comparativo de los sistemas estudiados

Entre los sistemas estudiados se encuentra SIGA e IBM Máximo los cuales brindan gran cúmulo de funcionalidades propias para cada una de las empresas para las que fueron creados. Ambos no resulta una solución factible para las entidades cubanas debido a que fueron desarrollados sobre plataformas de software propietario. Una de las debilidades del SIGA es que se centra en el cuidado a la información, dejando a un lado la protección del equipamiento tecnológico, que es hoy, uno de los principales problemas que presenta el centro; mientras que IBM Máximo no tiene implementado una forma de llevar a cabo el control de cada uno de los activos de una empresa ni registra las principales características de cada uno de ellos.

GLPI es una de las aplicaciones más utilizadas cuando se habla de Gestión de inventario, pero esta herramienta no gestiona el movimiento de los activos de un lugar a otro y es esto una de las necesidades que se desea solucionar hoy en el centro.

1.3.2- Gestión de incidentes de la seguridad de la información

La gestión de incidentes tiene como objetivo resolver de la manera más rápida y eficaz cualquier incidente que cause una interrupción en un servicio determinado (2). Entre los sistemas que se dedican a dichas actividades se encuentran los que se detallan a continuación.

1.3.2.1- GLPI

Una de las funcionalidades mejor desarrolladas de GLPI, es la gestión de solicitudes e incidencias que se realiza a través de una cómoda interfaz Web o mediante email. La aplicación permite asignar las incidencias a un técnico para su resolución, planificar las intervenciones y dejar incidencias en espera. Toda una gama de servicios que puede ser muy interesante para determinadas empresas. En este sentido se tiene la posibilidad de añadir costes por intervención, ya sea por horas, fijo o de material. Desde el punto de vista

⁴ VLAN: Red de área local virtual.

del técnico, las posibilidades abarcan todo lo necesario, desde gestión de prioridades, seguimiento e históricos, cambio de estado (apertura, resolución, pendiente, cierre), notificación por correo electrónico, gestión de tiempos y de costes (11).

1.3.2.2- Xperta: Gestor de incidentes

Xperta es una aplicación Web, los requisitos para lograr su funcionamiento son: un navegador Web y una conexión a Internet. Con la utilización de Xperta los usuarios tienen la facilidad de consultar el estado y los trabajos realizados por el servicio técnico en sus incidencias y permite tener un histórico de las mismas. A través de la interfaz los usuarios pueden informar de sus incidencias, sólo con introducir un título descriptivo de la incidencia y una breve descripción. De este modo los técnicos tendrán constancia desde este mismo momento de la incidencia (dentro del panel de administración de Xperta) y a través de un correo electrónico (si así está configurado en la aplicación). El sistema contiene una base de datos de todos los usuarios y personal del servicio técnico, de todas las incidencias actuales y solucionadas. Esto permite que los técnicos tengan acceso en todo momento a los datos del usuario o cliente (12).

1.3.2.3- Sistema de gestión de incidentes (UCI)

El Sistema de gestión de incidentes se encuentra en funcionamiento, se puede acceder a él mediante el sitio de Seguridad Informática de la UCI, este sitio no posee ningún tipo de referencia puesto que se consultó al personal que trabaja directamente con él y solo facilitaron información de las funcionalidades propias de la aplicación. El sistema permite el registro de las incidencia de redes, servicios telemáticos o relacionadas con seguridad informática. Estas incidencias se van registrando por orden de llegada en colas dependiendo de la clasificación estas pueden ser redes, servicios telemáticos o seguridad informática. Luego que se introduce la incidencia al sistema, esta le llega al técnico encargado de darle solución. Cuando el técnico tiene la respuesta a la misma, se le envía por email a la persona que registró la incidencia.

1.3.2.4- Análisis comparativo de los sistemas estudiados

Luego del análisis de los sistemas estudiados de gestión de incidencias y valorar cada uno de sus beneficios y dificultades se decide realizar un compendio de todos para la propuesta de solución pues el GLPI y el Xperta solo registran incidencia relacionadas con los activos tecnológicos mientras que el Sistema de gestión de incidentes (UCI) solo permite el registro de incidencia que estén relacionadas con redes, servicios telemático y seguridad informática. De cada uno se adoptan las principales funcionalidades y formas de registrar cada una de las incidencias y es por ello que el sistema que se espera desarrollar va a permitir el

registro de las incidencias tanto de los activos, como de seguridad informática y de las indisciplinas por parte del personal que visita los laboratorios de producción.

1.3.3- Seguridad Física y del Ambiente.

El tema de seguridad física y del ambiente está muy ligado con los sistemas de control de acceso que son esenciales para proteger la confidencialidad, integridad y disponibilidad de los activos más importantes de una organización, pues permiten que solo los usuarios autorizados accedan a los recursos que requieren, ya sean lógicos o físicos (2). Existen gran cantidad de tecnologías y mecanismos de control de acceso entre los que se encuentran la autenticación biométrica, las tarjetas inteligentes y los lectores de código de barra que a continuación se procede a realizar un estudio de cada uno de ellos.

1.3.3.1- Autenticación biométrica

La biometría, también conocida como técnicas de autenticación biométrica, surge con el objetivo de resolver el problema de la necesidad de autenticar de forma segura la identidad de las personas que pretenden acceder a un determinado servicio o recinto físico; a partir de características que son propias de cada individuo, como voz, huella dactilar, rostro, entre otras. Entre las ventajas de esta técnica se encuentra que los patrones no pueden perderse o ser sustraídos, ni pueden ser usados por otros individuos en el caso de que lleguen a tener accesible una tarjeta personal y/o PIN⁵. Así, por ejemplo, es habitual que en el caso de pérdida o sustracción de una cartera, cualquiera pueda hacerse pasar por uno mismo, ya que es muy frecuente tener junto a las tarjetas personales, el número secreto de la tarjeta apuntado. Este problema de suplantación de identidad quedaría resuelto con el uso de patrones biométricos como medio de identificación personal (13).

Los sistemas biométricos, se clasifican en:

- Fisiológicos: huella dactilar, iris, retina, cara, geometría de la mano, huella palmar, estructura de la venas, estructura de la oreja, termografía facial.
- Conductuales: voz, escritura, firma manuscrita, modo de teclear, modo de andar.

⁵ PIN: Personal Identification Number o número de identificación personal.

1.3.3.2- Tarjetas inteligentes

Distinto a una contraseña, un token⁶ de seguridad es un objeto físico, o sea, un pequeño dispositivo de hardware que los usuarios cargan consigo para autorizar el acceso a un servicio determinado. El dispositivo puede ser en forma de una tarjeta inteligente o puede estar incorporado en cualquier objeto. Los token proveen un nivel de seguridad adicional utilizando el método conocido como autenticación de dos factores: el usuario tiene un PIN, que le autoriza como el propietario del dispositivo; luego el dispositivo despliega un número que identifica en forma única al usuario ante el servicio, permitiéndole ingresar. Cuando el usuario poseedor de una tarjeta inteligente desea autenticarse, solo necesita introducir la tarjeta en un hardware lector; los dos dispositivos se identifican entre sí con un protocolo a dos bandas en el que es necesario que ambos conozcan la misma clave lo que elimina la posibilidad de utilizar tarjetas de terceros para autenticarse ante el lector de una determinada compañía; esta clave puede utilizarse para asegurar la comunicación entre la tarjeta y el dispositivo lector (7).

1.3.3.3- Lector de código de barra

El código de barras es una tecnología de identificación automática. Permite recolectar datos con precisión y rapidez. Un código de barras consiste en una serie de barras adyacentes paralelas y espacios. Los diseños predeterminados de anchura se utilizan para codificar datos en el código. Para leer información en un símbolo de código de barras, un dispositivo de lectura, tal como un lápiz óptico, se desliza a través del símbolo de un lado al otro, la anchura de barras y los espacios son analizados por el decodificador del lector, y los datos originales se recuperan. La aplicación más visible en esta tecnología es en la industria de supermercado, donde ha estado en el uso desde 1970 (14).

Los beneficios de utilizar códigos de barras como sistema de recolección de datos automática son muy sencillos: la velocidad y exactitud. Una y otra vez se ha probado que capturar datos con códigos de barras es por lo menos 100 veces más rápido y más exacto que la captura normal por teclado, que se traduce en un aumento dramático en la eficiencia y productividad para cualquier operación (14).

⁶ Token: mecanismo de control de acceso

1.3.3.4- Análisis comparativo de las tecnologías estudiadas

Luego de estudiadas algunas tecnologías existentes para el control de acceso, analizadas las ventajas y desventajas de las mismas, se arriba a la conclusión que ninguna de las mencionadas resuelven los objetivos de la propuesta de solución puesto que el centro no cuenta con la infraestructura tecnológica para su puesta en práctica. Se decide entonces a nivel de centro realizar el control de acceso mediante la entrada manual de los datos mediante el código de solapín de cada estudiante o trabajador.

1.4- Metodología de desarrollo de software

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar un software con calidad. Define las etapas para el desarrollo de un software, las tareas que se deben llevar a cabo en cada etapa, las restricciones que deben aplicarse, las técnicas y herramientas que deben emplearse, y la forma de controlar y gestionar un proyecto (15).

Existen metodologías de desarrollo de software ágiles y tradicionales, en la tabla que se muestra a continuación se representan cada una de sus características específicas (15):

Tabla I: Comparación entre metodologías ágiles y tradicionales (Luna Téllez, Linda; 2012)

Metodologías Ágiles	Metodologías Tradicionales
Preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas y normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado con el cliente.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños de 10 o menos integrantes y trabajan en el mismo sitio.	Grupos grandes y distribuidos.

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

Generan poca documentación.	Amplia gama de documentos.
Para proyectos pequeños y de corta duración.	Proyectos grandes y de largo proceso de vida.

Luego del estudio realizado y por las condiciones que presenta el proyecto de investigación se decide utilizar una metodología de desarrollo de software ágil pues es un proyecto pequeño, de corta duración, el equipo de desarrollo es de una persona y es necesaria la rápida puesta en marcha del producto.

Existe una amplia gama de metodologías de desarrollo de software ágiles entre las que se encuentran:

Programación Extrema (XP)

Representa una de las metodologías de desarrollo de software más exitosas. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, debido a que es uno de los requisitos para llegar al éxito del proyecto. Sus dificultades se centran en que es imposible prever todo antes de programar, no realiza un modelado del negocio ni un análisis y diseño de la propuesta de solución y no guarda una documentación donde en futuros cambios para el proyecto sea fácil de entender lo que se realizó (15).

Metodología AUP (Proceso Unificado Ágil)

AUP es una versión simplificada de Rational Unified Process (RUP). Este describe un enfoque simple y fácil de entender para desarrollar Software de aplicaciones de negocio usando técnicas y conceptos aunque aun permaneciendo como RUP. El enfoque aplica técnicas ágiles tales como: desarrollo manejado por las pruebas, gestión de cambios ágil, desarrollo ágil manejado por el modelo y rediseño de la Base de datos (16). Las disciplinas de AUP son diferentes a las de RUP, se han mezclado modelación del negocio, requisitos, análisis y diseño en la disciplina de modelación y se unieron gestión de cambios con gestión de configuración en una sola disciplina. En la Tabla II se representan cada una de las fases de desarrollo de AUP.

Tabla II: Fases de desarrollo de AUP (Metodología Ágil; 2013)

Fase	Objetivos	Hito
Inicio	Identificar el alcance inicial del proyecto, una arquitectura potencial y obtiene los fondos iniciales del proyecto y la aceptación de los involucrados.	Objetivos del ciclo de vida.
Elaboración	Probar la arquitectura del sistema.	Ciclo de vida de la arquitectura.
Construcción	Construir el Software que trabaje sobre bases iterativas y que encuentre las necesidades de mayor prioridad de los involucrados.	Capacidad operacional inicial.
Transición	Validar y desplegar el sistema en el ambiente de producción.	Entrega del producto.

Ventajas de AUP (16):

- Se apoya en un proceso formalizado como es RUP por lo que evita la improvisación.
- Tiene bien establecidos los roles y las fases.
- Es ágil y por tanto se basa en un proceso iterativo evolutivo.
- Incrementa la productividad.
- Facilita el trabajo de proyectos de pequeño tamaño.
- Hay información disponible de forma libre.

Luego del análisis realizado de las metodologías de desarrollo de software, se decide seleccionar AUP debido a todas las ventajas que proporciona, ya que permite reunir una amplia gama de documentación, un profundo y bien detallado modelado del negocio, análisis y diseño de la propuesta de solución, lo que admite tener un mayor conocimiento del desarrollo del sistema a futuros desarrolladores facilitando la etapa de mantenimiento de la solución.

1.5- Notación para el modelado del negocio

Entre las notaciones existentes para el modelado se encuentra Integrated Definition Modeling Language (IDEF) en su versión cero. Su objetivo fundamental es el modelado funcional de procesos, una de sus

funcionalidades es representar de manera estructurada y jerárquica las actividades que conforman un sistema o empresa y los objetos o datos que soportan la interacción de esas actividades. Está basada en un estándar con especificaciones precisas y rigurosas. Su principal deficiencia se centra en que no todas las herramientas de software soportan la notación IDEF0 (17).

La notación de modelado de procesos de negocio (BPMN) es un estándar para el modelado de procesos de negocio, y proporciona una notación gráfica para especificar los procesos de negocio en un diagrama de procesos de negocio (BPD) sobre la base de la técnica de diagrama de flujos muy similar a los diagramas de actividad de Lenguaje Unificado de Modelado (UML) (18).

Objetivos de BPMN:

- Resolver las dificultades de comunicación que tiene el lenguaje común.
- Apoyar la gestión de procesos empresariales, tanto para los usuarios técnicos como para los usuarios empresariales.

Características de BPMN (19):

- Proporciona un método normalizado para representar procesos de negocio.
- Facilita su entendimiento debido a la poca complejidad de su notación.
- Proporciona un lenguaje común entre los usuarios de negocio y los técnicos.
- Facilita la construcción de diagramas de los procesos de negocio.

Luego de analizadas las notaciones IDEF0 y BPMN se selecciona BPMN debido a que es una notación mucho más completa y brinda todas las funcionalidades necesarias para el desarrollo de la solución que se desea.

1.6- Marco de trabajo

Los marcos de trabajo son una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que pueden servir de base para la organización y desarrollo de un software (20).

1.6.1- Marcos de trabajo para PHP

Entre los marcos de trabajo más usados en el mundo se encuentra Symfony2, el popular framework⁷ para desarrollar aplicaciones sobre PHP 5.3 o superior. Se anunció por primera vez a principios de 2009 y constituye un cambio radical tanto en arquitectura interna como en filosofía de trabajo respecto a sus versiones anteriores. Symfony2 ha sido ideado para exprimir al límite todas las nuevas características de PHP 5.3 convirtiéndolo en uno de los frameworks PHP con mejor rendimiento. Su arquitectura interna está desacoplada, lo que permite reemplazar o eliminar muy fácil aquellas partes que no se ajustan al proyecto (20). Ofrece grandes ventajas en cuanto al aumento en la productividad, compatibilidad con varias plataformas operativas. Cuenta con varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web. Utilizando como patrón de arquitectura el Modelo Vista Controlador (MVC) y basado en las mejores ideas y prácticas de las principales tecnologías de desarrollo web actuales.

Zend Framework (ZF), es un marco de trabajo de código abierto para desarrollar aplicaciones web y servicios web con PHP5, es una implementación que usa 100% código orientado a objetos, la estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. La estructura de archivos no está definida por sí misma, cualquier desarrollador puede modificarla, lo que indica que no sería compatible con otra aplicación desarrollada con Zend, y requiere de una larga curva de aprendizaje (21).

Entre los marcos de trabajo que usan como lenguaje de desarrollo PHP se selecciona para el desarrollo de la aplicación Symfony en su versión 2.0, pues es el que más se ajusta a las características y condiciones del sistema que se desea desarrollar y proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web. Presenta una estructura muy desacoplada lo que permite seleccionar los bundle que se adapten al sistema que se desea desarrollar y de esta forma se le facilita el trabajo a los desarrolladores.

1.6.2- Librería JavaScript

ExtJS es una librería JavaScript que permite construir aplicaciones complejas en Internet pero al ser muy grandes, las descargas son lentas porque cargan todo al inicio hacen que sean más lentas que una

⁷ Framework: marco de trabajo.

aplicación web tradicional. Estas aplicaciones tienen problemas con los programas de accesibilidad pues, al igual que los motores de búsqueda, no trabajan bien con texto cargado de forma dinámica (22).

Otra de las librerías JavaScript estudiadas es JQuery, esta librería es rápida y concisa, simplifica la forma de interactuar con el documento HTML8, manejo de eventos, animación, y las interacciones Ajax para el desarrollo web rápido. JQuery es software libre y de código abierto, lo que significa que puede ser utilizado en proyectos libres. Entre sus principales ventajas se encuentran (23):

- Alterar el contenido de un documento.
- Responder a los eventos de un usuario.
- Recuperar información de un servidor sin refrescar una página.
- Animar cambios realizados a un documento.

Entre las librerías estudiadas se selecciona JQuery en su versión 1.7 pues es la que más se adecua a la solución que se desea desarrollar y presenta gran cantidad de información disponible para su rápido y fácil aprendizaje.

1.6.3- Mapeo Relacional de Objetos (ORM)

Entre los ORM existentes a nivel mundial uno de los más utilizados es Propel debido a que provee un sistema para almacenar objetos en una base de datos, su principal ventaja es que está integrado con Symfony y decenas de plugins solo funcionan para él, pero requiere de tiempo para su aprendizaje. Las aplicaciones son más lentas que con Doctrine esto es debido a que todas las consultas que se hagan sobre la base de datos, el sistema primero deberá transformarlas al lenguaje propio de la herramienta, luego leer los registros y por último crear los objetos. Otra de las desventajas es que en su versión 1.2 que se utiliza en la actualidad tiene un rendimiento muy mejorable (24).

Doctrine es un sistema ORM potente y completo para PHP 5.2 o mayor, con una capa de abstracción de la base de datos incorporada. Entre otras ventajas ofrece la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases a tablas de una base de datos. Su principal ventaja radica en poder acceder a la base de datos utilizando la programación orientada a objetos (POO). Doctrine utiliza el patrón Active Record para manejar la base de datos, tiene su

8 HyperText Markup Language o lenguaje de marcado hipertextual.

propio lenguaje de consultas de datos (DQL), trabaja de manera rápida y eficiente, y es fácil de integrarlo a los principales marcos de trabajo para el desarrollo utilizados en la actualidad (25).

Se escoge Doctrine en su versión 2.0 debido a que resulta más fácil para su aprendizaje, existe gran cantidad de documentación disponible que facilita su estudio y los generadores automáticos que posee permite facilitar el trabajo de mapeo de la base de dato a los desarrolladores.

1.6.4- Bootstrap

Bootstrap es un marco de trabajo desarrollado por Twitter que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. La mayor ventaja es que permite crear interfaces que se adapten a los distintos navegadores apoyándose en un marco de trabajo potente con numerosos componentes webs que ahorrarán mucho tiempo y esfuerzo.

Bootstrap ofrece una serie de plantillas CSS y ficheros JavaScript que permiten integrar el marco de trabajo de forma sencilla y potente en proyectos webs. Entre sus principales características están (26):

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tabletas y móviles a distintas escalas y resoluciones.
- Se integra con las principales librerías JavaScript, por ejemplo JQuery.
- Ofrece un diseño sólido y estándares como CSS3/HTML5.
- Dispone de distintos layout predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.

Para el decorado de las plantillas se usará Bootstrap en su versión 2.3 debido a que se adapta a diferentes navegadores y posee gran variedad de vistas muy agradables para el usuario.

1.7- Lenguajes utilizados

En este epígrafe se va a tratar acerca de los lenguajes utilizados para desarrollar la propuesta de solución. La selección de los mismos se realizó teniendo en cuenta los marcos de trabajo optados.

1.7.1- Lenguaje Unificado de Modelado (UML)

Es un lenguaje para el desarrollo de software orientado a objetos, su propósito es visualizar, especificar, construir y documentar proyectos de software. Ayuda al usuario a entender la realidad de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades en proyectos que no estén seguros en su desarrollo, reduciendo el coste y el tiempo empleado en la construcción de las piezas que

constituirán el modelo; posee una gran cantidad de propiedades que han sido las que han contribuido a hacer de UML el estándar de la industria que es en realidad, algunas de estas propiedades son (27):

- Modela estructuras complejas.
- Las estructuras más significativas que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- Utiliza operaciones abstractas como guía para variaciones futuras, incrementando variables si es necesario.

1.7.2- PHP 5.3

“PHP es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir a los desarrolladores la generación dinámica de páginas” (28). PHP es un lenguaje de programación interpretado, diseñado para la creación de páginas Web dinámicas y encargado de la interpretación del lado del servidor. Este lenguaje multiplataforma se interpreta y ejecuta de igual forma un script independientemente del tipo de plataforma (tipo de sistema operativo o tipo de servidor web) donde sea ejecutado (29).

Ventajas:

- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL, facilitando así el desarrollo de aplicaciones web dinámicas que acceden a bases de datos en tiempo real.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- No requiere definición de tipos de variables, se evalúan en tiempo de ejecución, es decir, con darles un valor ya queda definido también su tipo.
- Tiene manejo de excepciones.

1.7.3- Java Script

JavaScript es un lenguaje de programación web, se utiliza para desarrollo de sitios web. Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlo, al usarlo se

pueden crear páginas web dinámicas, menús desplegados, efectos visuales sencillos, manipular datos y crear aplicaciones web, utilizando poca memoria y manteniendo un tiempo de descarga rápido para tu página web. Los programas escritos con JavaScript se pueden probar en cualquier navegador sin necesidad de procesos intermedios (30).

1.7.4- HTML 5 (HyperText Markup Language)

HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Posee etiquetas para manejar grandes conjuntos de datos permitiendo así generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido. Propone mejoras en los formularios y nuevos tipos de datos que facilitan validar el contenido (31).

1.7.5- Css 3 (Cascading Style Sheets)

Las hojas de estilo en cascada hacen referencia a un lenguaje usado para describir la presentación semántica de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML. CSS tiene una sintaxis muy sencilla, usa palabras claves para especificar los nombres de sus selectores, propiedades y atributos (32).

1.8- Herramientas seleccionadas para la solución

1.8.1- Herramientas CASE

Rational Rose Enterprise Edition es una herramienta CASE⁹ desarrollada por Rational Corporation basada en UML, que permite crear los diagramas que se van generando durante el proceso de Ingeniería en el Desarrollo del Software. Esta herramienta permite que el equipo de desarrollo entienda mejor el problema, que identifique las necesidades del cliente en forma más efectiva y comunique la solución propuesta de forma más clara. Rational permite completar una gran parte de las disciplinas de RUP (33).

Entre las características principales de Rational se pueden destacar que admite como notaciones UML y OMT¹⁰, permite desarrollo multiusuario, genera documentación del sistema y disponible en múltiples plataformas (33).

⁹ CASE: Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora

¹⁰ OMT: Object Modeling Technique

Visual Paradigm for UML es una herramienta flexible ya que es multiplataforma, es decir, tiene la capacidad de ejecutarse sobre diferentes sistemas operativos. Soporta todo el ciclo de vida del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Como lenguaje de modelado utiliza UML y apoya la última versión del estándar BPMN, ofreciendo soluciones de software que permiten a las organizaciones desarrollar las aplicaciones con mayor calidad, de forma satisfactoria y más barata. Posee gran facilidad de uso y el entorno gráfico que brinda es agradable para los usuarios. Presenta una buena interoperabilidad con otras herramientas CASE y con los principales IDE¹¹ (34).

Se selecciona para el desarrollo de la propuesta de solución Visual Paradigm en su versión 8.0 porque soporta aplicaciones web y escritorio, genera código y la base de dato a partir de los diagramas UML, permite la generación de informes en diferentes formatos, provee abundantes tutoriales posee gran robustez, usabilidad y portabilidad.

1.8.2- Entorno de desarrollo

Un IDE consiste en un programa informático que ha sido instalado en la máquina del desarrollador y cuyo objetivo es la elaboración de otras aplicaciones, es decir, es un entorno de programación que ha sido empaquetado como “Software”. En la actualidad existen gran cantidad de IDE de desarrollo entre los más usado se encuentra Eclipse y NetBeans.

Eclipse es un IDE de código abierto, multiplataforma, provee herramientas para administrar áreas de trabajo, construir, lanzar y depurar aplicaciones, permite compartir artefactos con un equipo y versionar el código fuente. Ha sido utilizado para construir ambientes para un amplio rango de temas como el desarrollo en lenguaje de programación Java, Servicios Web y certámenes de programación de juegos (35).

Por otra parte NetBeans es un producto libre y gratuito sin restricciones de uso. Está bajo las licencias Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL). NetBeans es una herramienta para programadores utilizada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación, posee un motor para escribir (resaltando la sintaxis), para identificar errores y para depurar el código. Sin lugar a dudas NetBeans se ha convertido en un IDE apto para la mayoría de los lenguajes de programación de código abierto modernos (36).

¹¹ IDE: Integrated Development Environment.

Por todo lo expuesto se selecciona NetBeans como IDE de desarrollo de la aplicación porque presenta una gran cantidad de ventajas y facilidades para el programador, proporciona características muy útiles como: gestor de ventanas, API (Interfaz de programación de aplicaciones) de acciones, API para la creación de diálogos, y la integración con la ayuda de java.

1.8.3- Servidor de aplicaciones

Tipo de servidor que permite el procesamiento de datos de una aplicación de cliente. Las principales ventajas de la tecnología de los servidores de aplicación es la centralización y la disminución de la complejidad del desarrollo de aplicaciones, dado que las aplicaciones no necesitan ser programadas; en su lugar, estas son ensambladas desde bloques provistos por el propio servidor (37).

Apache 2.2 es un servidor web de software libre, cuyo objetivo es servir o suministrar páginas web (en general, hipertextos) a los clientes web o navegadores que las solicitan, funciona sobre cualquier plataforma, permite que otros ordenadores vean la Web mediante un navegador. Es una solución altamente configurable y extensible a través de módulos, se integra con varias tecnologías, lenguajes, plataformas y bases de datos (37).

1.8.4- Sistemas gestores de base de datos

Un sistema gestor de base de datos como su nombre lo indica facilita la gestión de las bases de datos, en otras palabras, es una aplicación que actúa como mediador en forma de interfaz entre el usuario y la base de datos facilitando la creación, uso y administración de la misma. En la actualidad existen gran cantidad de gestores entre los que se encuentran MySQL y PostgreSQL.

Entre las principales características de MySQL se encuentran que aprovecha la potencia de sistemas multiprocesadores, soporta gran cantidad de tipos de datos para las columnas, dispone de API en gran cantidad de lenguajes, posee gran portabilidad entre sistemas y permite la gestión de usuarios y contraseñas, manteniendo buen nivel de seguridad en los datos. Pero no soporta transacciones, ni subselects¹² y no considera las claves ajenas, ignora la integridad referencial, dejándola en manos del programador de la aplicación (38).

¹² Subselec: es un comando SELECT dentro de otro comando es decir una consulta dentro de otra.

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multiversión, soportando todas la sintaxis SQL (incluyendo sub consultas, transacciones, tipos y funciones definidas por el usuario). Entre sus características principales se encuentran la integridad referencial, que posee procedimientos almacenados y funciones de compatibilidad para ayudar en la transición desde otros sistemas menos compatibles con SQL (34).

Posee un grupo de ventajas entre las que se encuentran una instalación ilimitada, posee mejor soporte que los proveedores comerciales, ahorros considerables en costos de operación, estabilidad y confiabilidad legendarias, es extensible, multiplataforma, y está diseñado para ambientes de alto volumen.

Para el sistema a implementar se selecciona la utilización de Postgre SQL en su versión 9.1 porque es un sistema de gestión de base de datos relacional orientado a objetos de software libre, que puede ser utilizado en los principales sistemas operativos: Linux y Windows.

1.9- Conclusiones parciales

El estudio de la Norma Cubana ISO/IEC 17799: 2007 posibilitó definir los procesos y las cláusulas por las cuales regir la presente investigación, las cuales son: la gestión de activos, la gestión de incidentes y la seguridad física o del ambiente. Se realizó un análisis de los sistemas y tecnologías existentes tanto en Cuba como en el resto del mundo lo cual arrojó como resultado que ninguno satisface a plenitud las necesidades que posee CEGEL, es por ello que se propone implementar un sistema de gestión de los laboratorios para satisfacer las demandas y peticiones que existen en el centro. Se define AUP como metodología de desarrollo de software para la propuesta de solución pues permite el desarrollo de aplicaciones en corta duración y propone un profundo análisis y diseño. En este capítulo se seleccionan las herramientas, tecnologías y lenguajes a tener en cuenta en los procesos de análisis, diseño e implementación del producto.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.1- Introducción

En el presente capítulo se describen los procesos del negocio relacionados con el objeto de estudio, se realizan los mapas de procesos que describen las principales actividades que se realizan para la Gestión de laboratorios de CEGEL. Se identifican los requisitos funcionales a partir del uso de técnicas, se especifican y definen los requisitos no funcionales. Se elabora el modelo conceptual con los principales conceptos que se manejan. Se realizan los diagramas definidos por la metodología utilizada y se brinda una detallada descripción de la arquitectura utilizada, de los patrones de diseño utilizados y se definen los estándares de codificación utilizados en el desarrollo de la aplicación.

2.2- Modelado de negocio

Una de las actividades realizadas en el modelado del negocio de la presente investigación fue la confección de los diagramas de procesos del negocio y la descripción de cada uno de ellos. Un diagrama de proceso contribuye hacer visible los procesos que componen un sistema, y sus relaciones principales. Permite identificar los individuos que intervienen en el proceso, la tarea que realizan y el valor de cada tarea o su contribución al proceso (39). A continuación se muestra la descripción y el diagrama de proceso de negocio correspondiente a cada una de las actividades realizadas por los técnicos generales en el centro CEGEL.

2.2.1.- Descripción del proceso Control de acceso de personas a locales

El proceso comienza cuando se desea acceder al laboratorio, en ese momento el técnico registra en un modelo de control los datos personales y la hora de entrada del individuo, es responsabilidad del técnico registrar a cada una de las personas que accede al laboratorio. En el momento de la salida el técnico solo busca a la persona en el modelo de control y escribe el horario de salida concluyendo así el proceso.

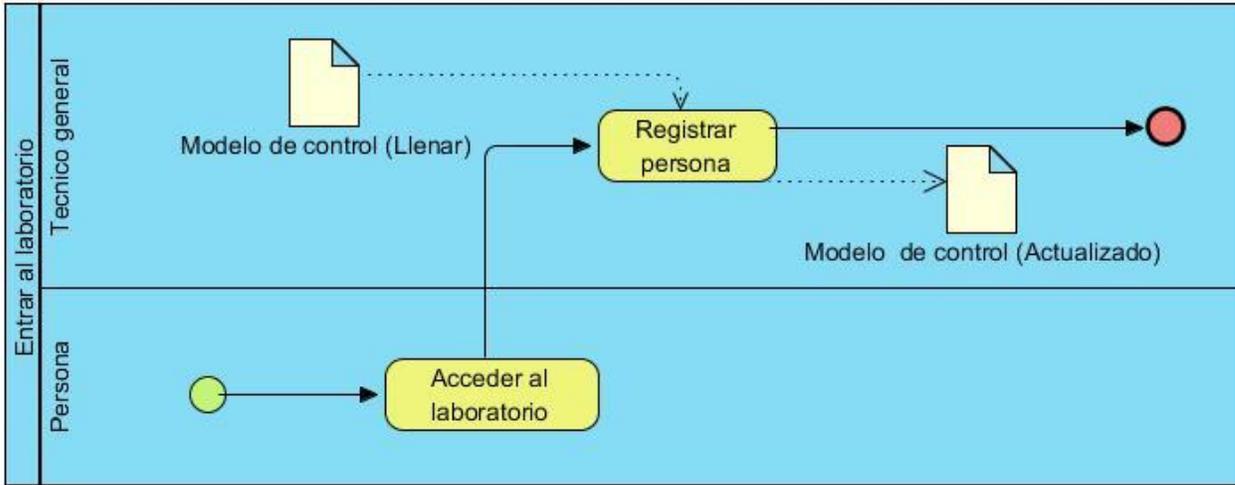


Figura 1: Diagrama del proceso Entrar al laboratorio

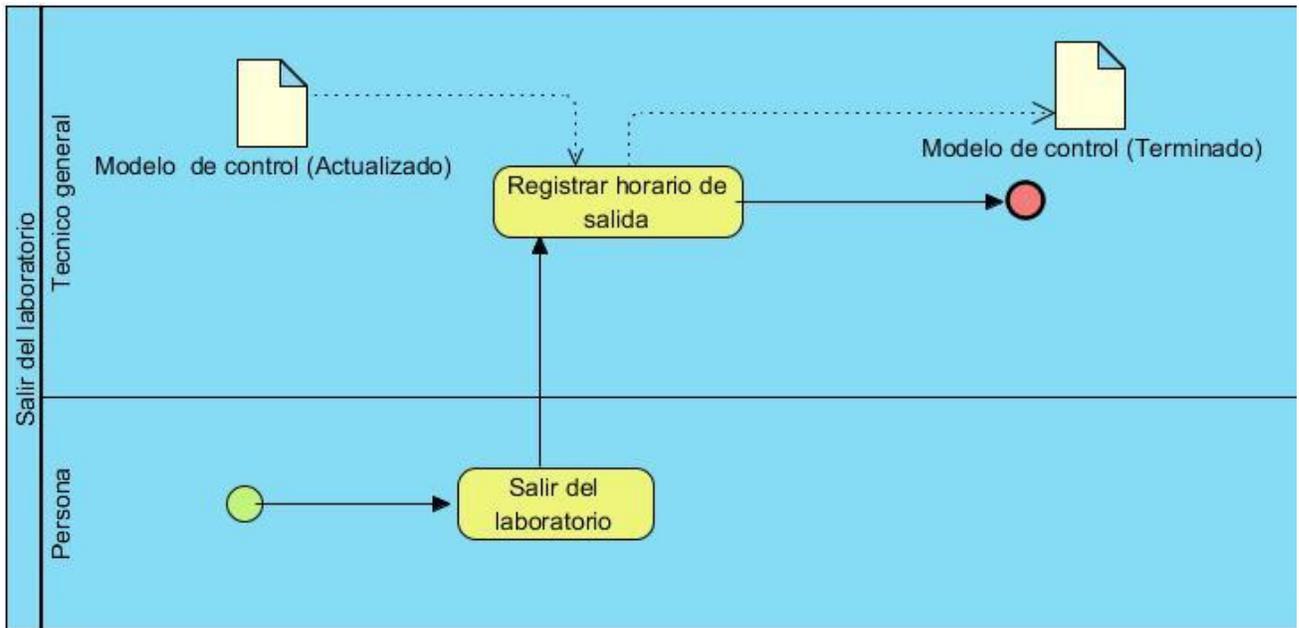


Figura 2: Diagrama del proceso Salir del laboratorio

2.2.2- Descripción del proceso Registro de incidencias

Este proceso comienza cuando se detecta una incidencia de cualquier tipo dentro de los laboratorios de producción, esto incluye detección de virus, roturas de equipos tecnológicos o pérdida de alguno de ellos o cualquier tipo de incumplimiento por parte de estudiantes, profesores o especialistas. Luego de detectada

se procede a registrarla en un modelo para el registro de incidencias, a continuación se notifica al jefe administrativo y así se termina la actividad.

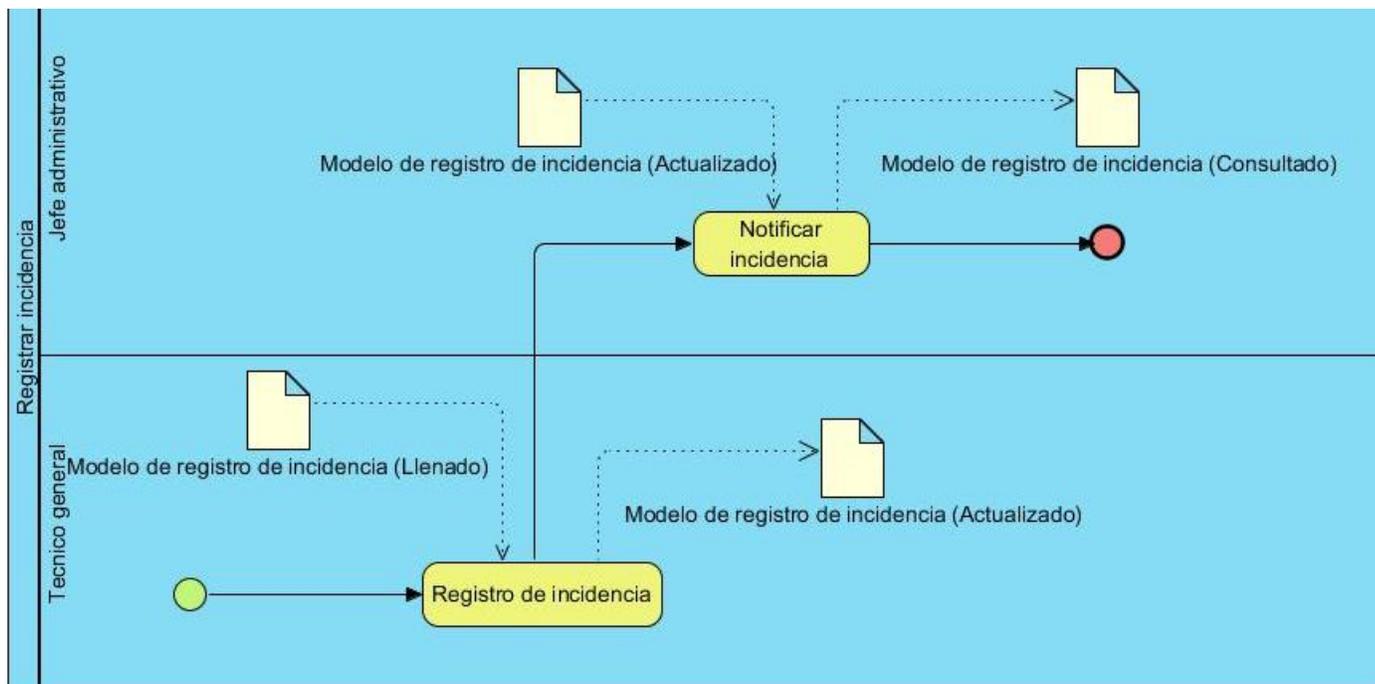


Figura 3: Diagrama del proceso Registro de incidencia

2.2.3- Descripción del proceso Recepción y entrega de medios

Este proceso comienza cuando se va a realizar la entrega o recepción de algún laboratorio. Las tareas básicas a cumplir con las siguientes: primero se realizar el conteo físico de los equipos tecnológicos, luego se procede a realizar el chequeo de los componentes internos de cada equipo, seguido se chequea el estado de los sellos de las PC, sobre todo aquellas que están rotas que no puedan chequearse su funcionamiento, ante cualquier irregularidad se realiza un reporte de incidencia. Luego se procede a confeccionar el modelo de entrega y recepción de los laboratorios.

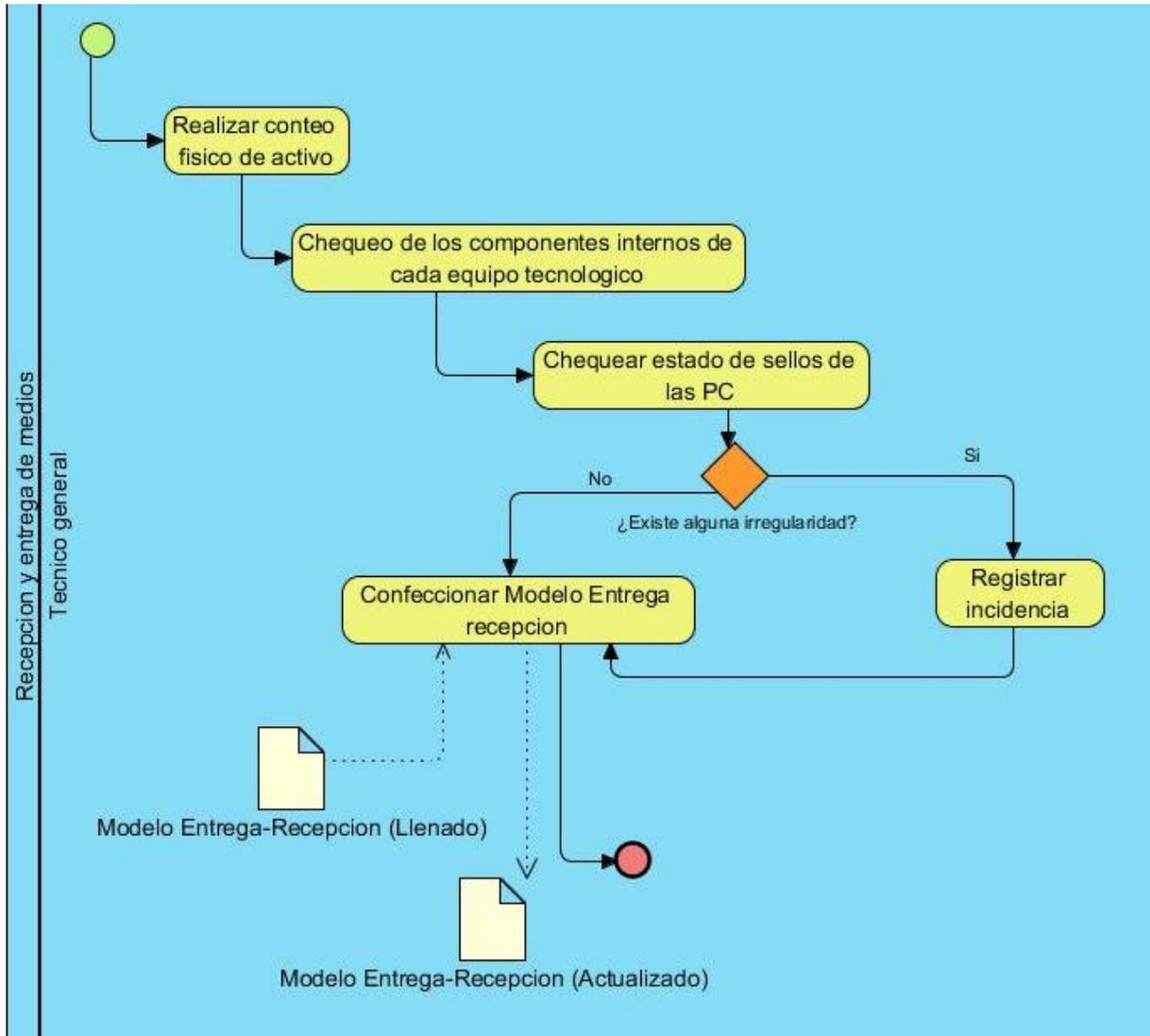


Figura 4: Diagrama del proceso Recepción y entrega de medios

2.3- Requisitos

Un requisito puede definirse como un atributo necesario dentro de un sistema, que puede representar una capacidad, una característica o un factor de calidad del sistema de tal manera que le sea útil a los clientes o a los usuarios finales (40).

2.3.1- Técnicas para la captura de requisitos

Se aplicaron las siguientes técnicas para la captura de los requisitos:

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

Entrevista: a través de encuentros planificados con el cliente se realizaron preguntas relacionadas con el proceso de Gestión de laboratorios de CEGEL en función de obtener las necesidades que tenían de informatización para la mejora de dicho proceso. Esto proporcionó como resultado que fueran identificados los requisitos por los que se guiaría el desarrollo de la aplicación.

Tormenta de ideas: en cada encuentro fueron debatidos los requisitos identificados, fluyendo variadas ideas acerca de los mismos, lo cual arrojó como resultado que se obtuviera una visión más amplia de lo que se quería implementar, favoreciéndose de esta forma el avance de futuras etapas en el desarrollo de la aplicación.

2.3.2- Especificaciones de los requisitos funcionales

Los requisitos funcionales describen la funcionalidad o los servicios que se espera que el sistema provea, sus entradas, salidas, excepciones, entre otras. Los requisitos funcionales identificados para el desarrollo de la aplicación son un total de 54 de ellos se tienen con complejidad alta 35, con media 10 y con baja 9. A continuación se muestra cada requisito, una breve descripción y la complejidad.

Tabla III: Descripción de requisitos funcionales

No.	Nombre	Descripción	Complejidad
RF #1	Autenticar usuario	El sistema permite la autenticación de un usuario.	Alta
RF #2	Realizar búsquedas	El sistema permite realizar búsquedas en la aplicación.	Alta
RF #3	Registrar incidencia	El sistema permite el registro de todas las incidencias que se sucedan durante el turno.	Alta
RF #4	Generar incidencia	El sistema genera la incidencia, en el caso de que haya problemas en la entrega se hace de manera automática.	Alta
RF #5	Visualizar incidencia	El sistema visualiza la incidencia.	Baja

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

RF #6	Registrar persona interna	El sistema permite registrar una persona interna.	Alta
RF #7	Modificar persona interna	El sistema permite modificar los datos de una persona interna.	Alta
RF #8	Visualizar persona interna	El sistema permite visualizar los datos de una persona interna.	Baja
RF #9	Eliminar persona interna	El sistema permite eliminar los datos de una persona interna.	Media
RF #10	Realizar entrega	El sistema permite registrar los datos de entrega de los laboratorios.	Alta
RF #11	Verificar entrega	El sistema verifica los datos introducidos con los que deberían según los datos que tiene el laboratorio, en caso de haber algún problema se genera una incidencia.	Alta
RF #12	Modificar entrega	El sistema modifica la entrega en el acto en que se crea, una vez que se dé listo a la entrega, no se puede modificar más.	Alta
RF #13	Acceder al laboratorio	Es cuando se le introduce al sistema el número de solapín.	Alta
RF #14	Registrar laboratorio	El sistema permite registrar o crear un nuevo laboratorio.	Alta
RF #15	Modificar laboratorio	El sistema permite modificar los datos de un laboratorio.	Alta
RF #16	Eliminar laboratorio	El sistema permite eliminar un laboratorio.	Media

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

RF #17	Visualizar laboratorio	El sistema visualiza o muestra los laboratorios existentes.	Baja
RF #18	Asignar activos a los laboratorios	El sistema permite asignar los activos a cada laboratorio.	Alta
RF #19	Asignar personas a los laboratorios	El sistema permite asignar las personas a cada laboratorio.	Alta
RF #20	Asignar técnicos a las áreas	El sistema permite asignar técnicos por cada una de las áreas.	Alta
RF #21	Adicionar nomenclador tipo incidencia	El sistema permite registrar o crear un nuevo tipo de incidencia.	Alta
RF #22	Modificar nomenclador tipo incidencia	El sistema permite registrar o crear un nuevo tipo de incidencia.	Alta
RF #23	Eliminar nomenclador tipo incidencia	El sistema permite eliminar un tipo de incidencia.	Media
RF #24	Visualizar nomenclador tipo incidencia	El sistema visualiza o muestra los nomencladores tipo incidencia.	Baja
RF #25	Adicionar nomenclador tipo activo	El sistema permite registrar o crear un nuevo tipo activo.	Alta
RF #26	Modificar nomenclador tipo activo	El sistema permite registrar o crear un nuevo tipo activo.	Alta
RF #27	Eliminar nomenclador tipo activo	El sistema permite eliminar un tipo activo.	Media

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

RF #28	Visualizar tipo activo	nomenclador	El sistema visualiza o muestra los nomencladores tipo activo.	Baja
RF #29	Adicionar proyecto	nomenclador	El sistema permite registrar o crear un nuevo proyecto.	Alta
RF #30	Modificar proyecto	nomenclador	El sistema permite registrar o crear un nuevo proyecto.	Alta
RF #31	Eliminar proyecto	nomenclador	El sistema permite eliminar un proyecto	Media
RF #32	Visualizar proyecto	nomenclador	El sistema visualiza o muestra los nomencladores proyecto.	Baja
RF #33	Adicionar área	nomenclador	El sistema permite registrar o crear una nueva área.	Alta
RF #34	Modificar área	nomenclador	El sistema permite registrar o crear una nueva área.	Alta
RF #35	Eliminar nomenclador área		El sistema permite eliminar un área.	Media
RF #36	Visualizar área	nomenclador	El sistema visualiza o muestra los nomencladores área.	Baja
RF #37	Reporte de incidencia		El sistema genera un reporte con las incidencias ocurridas durante el turno.	Alta
RF #38	Generar reportes		El sistema permite generar reportes de forma automática.	Alta
RF #39	Adicionar activo		El sistema permite adicionar un nuevo activo.	Alta

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

RF #40	Modificar activo	El sistema permite modificar un activo.	Alta
RF #41	Eliminar activo	El sistema permite eliminar un activo.	Media
RF #42	Adicionar usuario	El sistema permite adicionar un usuario al sistema.	Alta
RF #43	Modificar usuario	El sistema permite modificar un usuario del sistema.	Alta
RF #44	Eliminar usuario	El sistema permite eliminar un usuario.	Media
RF #45	Registrar persona externa	El sistema permite registrar una persona externa.	Alta
RF #46	Modificar persona externa	El sistema permite modificar los datos de una persona externa.	Alta
RF #47	Visualizar persona externa	El sistema permite visualizar los datos de una persona interna.	Baja
RF #48	Eliminar persona externa	El sistema permite eliminar los datos de una persona externa.	Media
RF #49	Envío de correos	<p>El sistema envía correos cuando se registra o modifica una incidencia de un tipo determinado a los usuario que tengan los roles que se definieron cuando se creó el tipo de incidencia.</p> <p>También se envía correo cuando se realiza la entrega de laboratorio y hay una incongruencia con lo que se debía entregar en este caso se le</p>	Alta

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

			manda el correo a los jefes administrativos y las personas que reciben y entregan el laboratorio.	
RF #50	Adicionar centro	nomenclador	El sistema permite adicionar un nuevo centro.	Alta
RF #51	Modificar centro	nomenclador	El sistema permite modificar un centro ya creado.	Alta
RF #52	Eliminar centro	nomenclador	El sistema permite eliminar un centro	Media
RF #53	Visualizar centro	nomenclador	El sistema permite visualizar los centros creados	Baja
RF #54	Asignar roles a los usuario		El sistema permite asignar roles a los usuarios.	Alta

2.3.3- Especificaciones de los requisitos no funcionales

Los requisitos no funcionales, son aquellos donde no se refieren a las funciones específicas que proporcionará el sistema, sino a las propiedades emergentes de este como: fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. El sistema a desarrollar debe cumplir con los siguientes requisitos no funcionales.

Apariencia o Interfaz Externa

Debe mostrar una interfaz gráfica agradable al usuario, un diseño sencillo para que el usuario realice cualquier tipo de interacción con el sistema de manera fácil. El producto de software final debe diseñarse sobre una arquitectura web.

Usabilidad

El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras.

Portabilidad

El sistema será un sistema multiplataforma pero en un inicio será desplegado en un ambiente Linux por políticas de la universidad.

Seguridad

Confiabledad: los datos de los usuarios registrados, no pueden ser utilizados para fines personales; en tal sentido y con el objetivo de proteger la información, se propone establecer un nivel de acceso a la misma mediante usuarios los cuales tendrán roles asociados.

Integridad: la información que es manejada por el sistema permanecerá inalterada a menos que sea modificada por el personal autorizado.

Disponibilidad: la aplicación debe de estar disponible en todo momento.

Software

Sistema operativo: Linux y Windows.

Ambiente de ejecución

Para el cliente:

- Navegador Mozilla Firefox versión 18.0.1 o superior.

Para el servidor:

- Servidor web Apache 2.2
- PostgreSQL 9.1

Hardware

Para el cliente:

- Tarjeta de red.
- Memoria RAM de 512Mb como mínimo.

Para el servidor:

1. Memoria RAM de 2Gb como mínimo.
2. Disco Duro de 250Gb.
3. Procesador Pentium 4 a 2.00 GHz.

2.4 -Análisis

2.4.1- Modelo conceptual

El modelo conceptual es una representación del sistema mediante definiciones organizadas en forma estructurada, está conformado por conceptos que son relevantes para el área que se analiza, así como la relación entre ellos. El modelo conceptual del proceso de Gestión de los laboratorios de CEGEL fue confeccionado con todas las clases conceptuales identificadas en este proceso y las relaciones existentes entre las mismas.

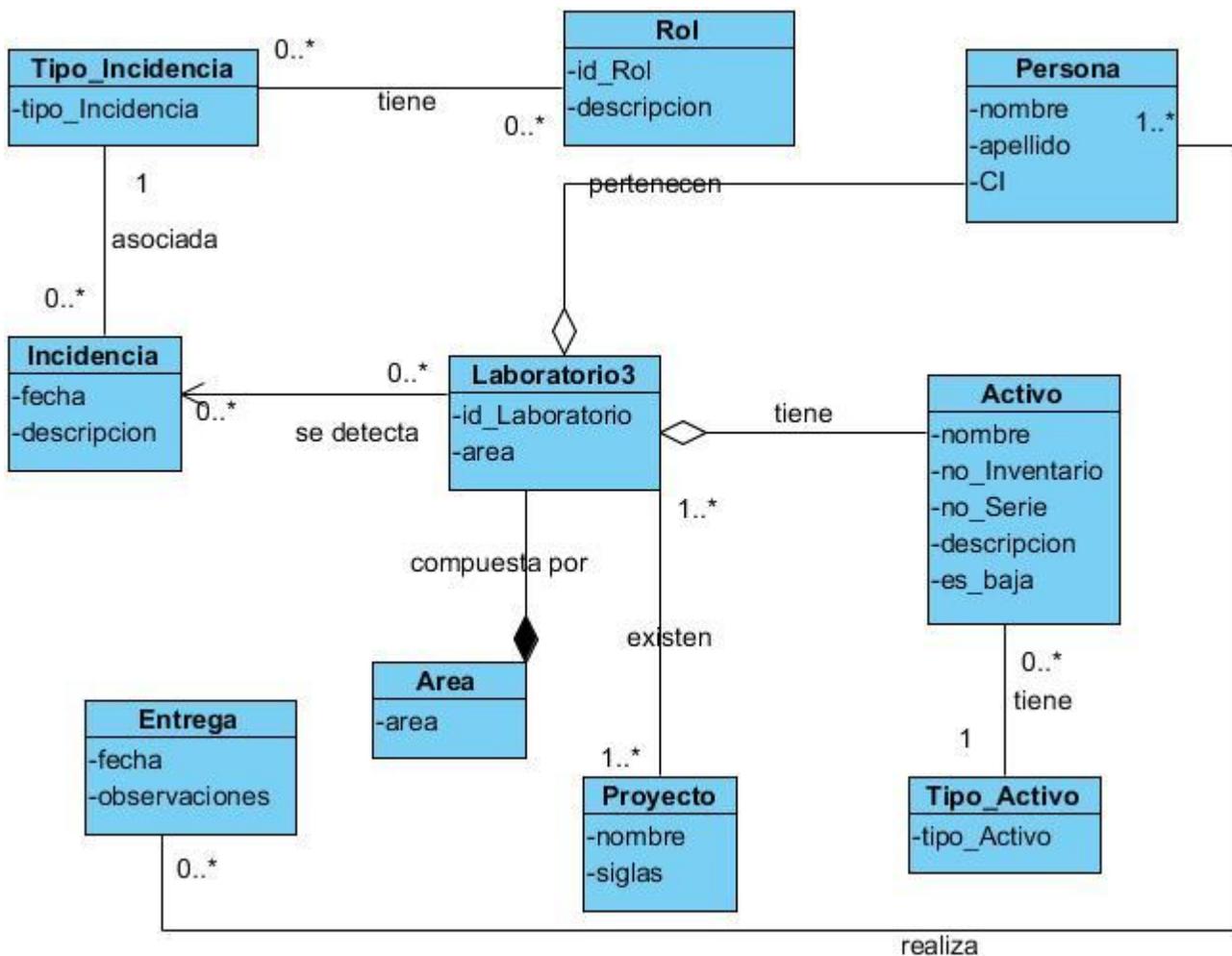


Figura 5: Modelo conceptual

2.4.2- Modelo de Caso de Uso

El Modelo de Casos de Uso describe las funcionalidades de un sistema. Un caso de uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema, estos se utilizan para la comunicación clara y el levantamiento de requisitos del desarrollo de un sistema, representándolos mediante un Diagrama de Casos de Uso. También se describen los actores del sistema y los casos de uso que van a representar las funcionalidades.

2.4.2.1-Determinación y justificación de los actores del sistema

Tabla IV: Actores y descripción

Actor	Justificación
Usuario	Es la persona que de una forma u otra interactúa con la aplicación (profesores, técnicos de laboratorios, personal de la dirección).
Técnico de Laboratorio	Es la persona que se encarga del registro del personal que accede a los laboratorios, del proceso de entrega y recepción de medios y del registro de incidencia.
Administrador	Es aquella persona responsable del correcto funcionamiento de la aplicación y el encargado de asignar roles y permiso a cada usuario del sistema.
Jefe Administrativo	Es la persona (Jefe de Centro, Jefe de Departamento de Tecnología, Decano, Rectora, entre otros.) que su interés es ver el registro de las incidencia y generar un reporte por alguna causa determinada.

2.4.2.2- Diagrama de Caso de Uso del Sistema

Los Diagramas de Casos de Uso del sistema son un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado observable para un actor. A continuación se muestra la figura correspondiente al Diagrama de Casos de Uso del Sistema.

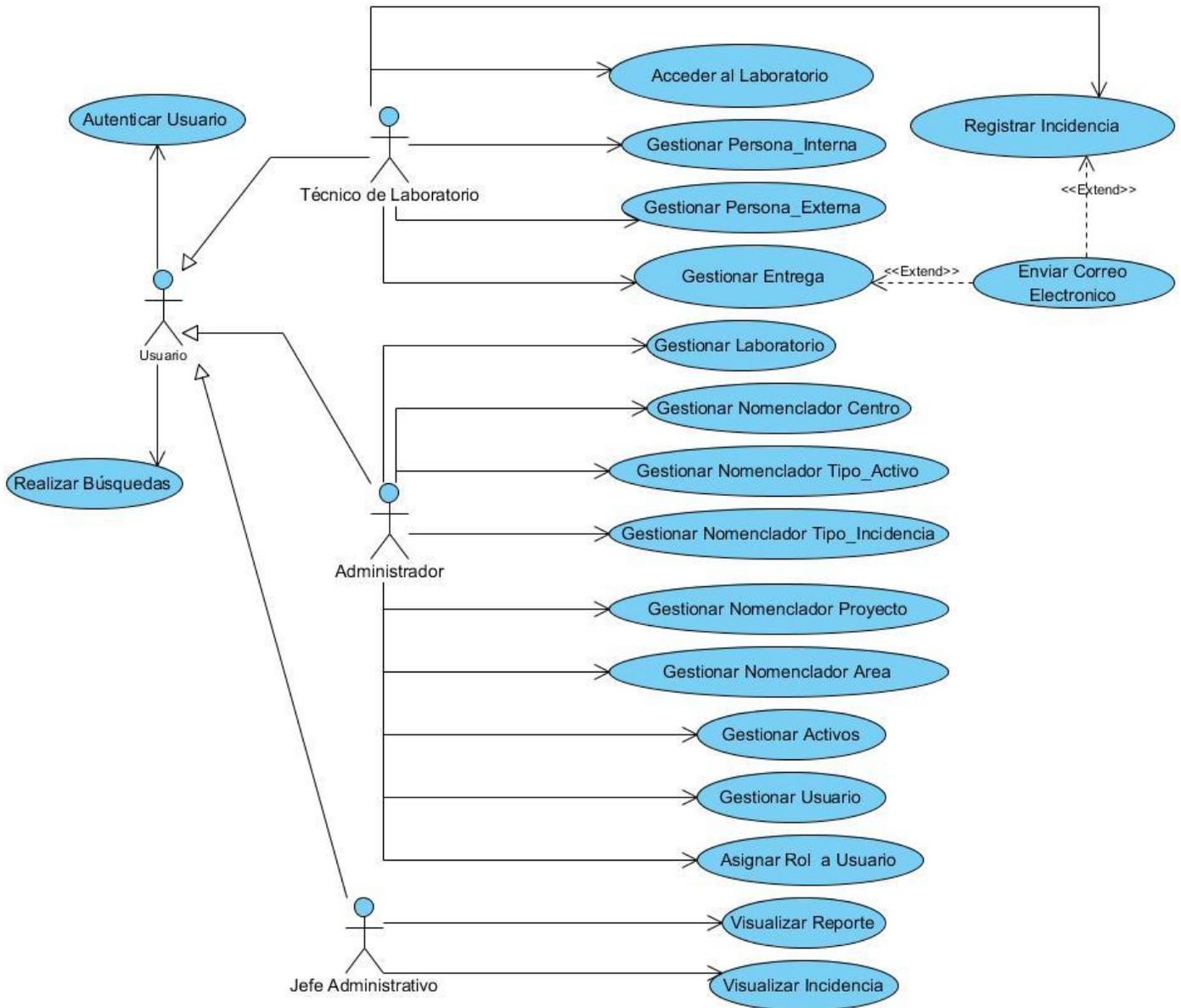


Figura 6: Diagrama de caso de uso del sistema

2.4.2.3- Descripción de los Casos de Uso del Sistema

La Descripción Textual de los Casos de Uso del Sistema describe bajo la forma de acciones y reacciones el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema (Consultar documento entregable Especificación de Casos de Uso) y ver Anexo# 1. En la Tabla V se

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

mencionan cada uno de los casos de uso de la aplicación los cuales reúnen todas las funcionalidades del sistema.

Tabla V: Casos de uso y complejidad

Ref. #	Caso de Uso	Complejidad
CU#1	Acceder al Laboratorio	Alta
CU#2	Realizar Búsquedas	Alta
CU#3	Registrar Incidencias	Alta
CU#4	Visualizar Incidencia	Media
CU#5	Gestionar Personas Interna	Alta
CU#6	Autenticar Usuario	Alta
CU#7	Gestionar Rol a usuario	Alta
CU#8	Gestionar Laboratorios	Alta
CU#9	Visualizar Reportes	Media
CU#10	Gestionar Nomenclador Centro	Alta
CU#11	Gestionar Entrega	Alta
CU#12	Gestionar Nomenclador Proyecto	Alta
CU#13	Gestionar Nomenclador Área	Alta
CU#14	Gestionar Nomenclador Tipo_Activo	Alta
CU#15	Gestionar Nomenclador Tipo_Incidencia.	Alta

CU#16	Gestionar Personas Externa	Alta
CU #17	Gestionar Activos	Alta
CU #18	Gestionar Usuario	Alta
CU #19	Enviar Correo Electrónico	Alta

2.5- Diseño

2.5.1- Descripción de la Arquitectura

El sistema se implementa sobre una arquitectura en capas la cual proporciona una organización jerárquica, donde cada capa proporciona servicios a la capa inmediata superior y se sirve de las prestaciones que le brinda la inmediata inferior.



Figura 7: Flujo de la información entre las capas

El estilo soporta un diseño basado en niveles de abstracción crecientes lo cual a su vez le permite a los implementadores dividir un problema complejo en una secuencia de pasos incrementales. Se pueden realizar optimizaciones y refinamientos enfocando los cambios en un solo lugar. Proporciona amplia reutilización dada la división bien definida de responsabilidades. Una especialización muy usada de la arquitectura en capas es la arquitectura de tres capas donde se observan muy bien delimitadas las responsabilidades de cada funcionalidad en la aplicación: presentación, negocio y datos. A continuación se muestra una breve descripción de cada una de las capas de la arquitectura definida.

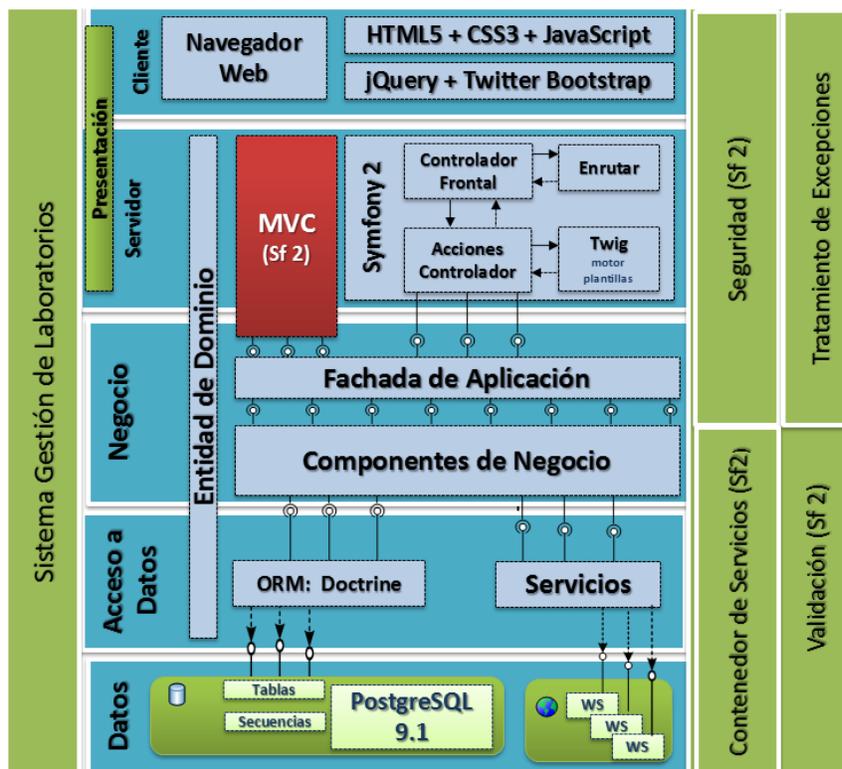


Figura8: Esquema de la arquitectura del sistema

2.5.1.1- Capa de Presentación

Esta capa es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa contiene los componentes que implementan y muestran la interfaz de usuario (UI), administra la interacción con el usuario y ejecuta la lógica asociada con la validación de los datos. Los componentes de UI proporcionan una forma para que los usuarios interactúen con la aplicación, mostrando los datos en el formato adecuado. Asimismo, obtienen y validan los datos entrados por el usuario.

La capa de presentación es una sola, pero atendiendo a su distribución física se puede ver dividida en dos: del lado del cliente y del lado del servidor. Del lado del cliente la capa de presentación realizará la interacción con el usuario mediante componentes de entrada y salida de información y componentes de control de procesos de interfaz de usuario. Estos componentes de control de UI serán escritos en Java Script y

manipulados a través del marco de trabajo jQuery el cual realizará la lógica asociada a la validación de los datos proporcionados por el usuario.

Del lado del servidor la capa de presentación será gestionada usando el motor de plantillas Twig. Como medida de seguridad siempre se realizará la validación de los datos en el servidor, independientemente a la validación hecha en el cliente.

```
class LaboratorioController extends BaseController
{
    public function newAction()
    {
        $entity = new Laboratorio();
        $form = $this->createForm(new LaboratorioType(), $entity);

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

Figura 9: Ejemplo de clase controladora creando clases de la vista

En Esta capa también se implementa el patrón Modelo Vista Controlador (MVC) que se explica a continuación.

Modelo Vista Controlador (MVC)

Este patrón de arquitectura de software se encarga de separar los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El modelo representa los datos y las reglas de negocio que rigen su acceso y actualización; puede verse como una modelación de los procesos del mundo real.

Las vistas se encargan de presentar los datos obtenidos del modelo. Es responsabilidad de las vistas mantener la información actualizada, esto se puede lograr a través de peticiones de actualización al modelo o a través de notificaciones de cambio que el modelo emite (eventos).

El controlador actúa como un traductor de las acciones que se realizan en las vistas en las operaciones que ocurren en el modelo. Las acciones realizadas por el modelo desencadenan la activación de procesos de

negocio o cambian el estado del modelo. Sobre la base de las acciones del usuario y los resultados del modelo, el controlador responde mediante la selección de la vista apropiada.

2.5.1.2- Capa de Negocio

La capa de negocio es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso de negocio. Se denomina capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

```
use Base\ArquitecturaBundle\Negocio\Gestor\BaseGtr;  
  
class LaboratorioGtr extends BaseGtr {  
  
    /** [...]   
    public function getLaboratorios() [...]   
  
    /** [...]   
    public function getLaboratorioDadoId($id) [...]   
  
    /** [...]   
    public function getPersonasInternas($id) [...]   
}
```

Figura 10: Ejemplo de clase gestora del negocio

2.5.1.3- Capa de Acceso a Datos

La capa de Acceso a Datos es la encargada de brindar los mecanismos necesarios para acceder a los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, puede estar compuesta por fuentes de datos como dispositivos y servicios web, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

2.5.1.4- Capa de datos

La capa de datos es donde residen los datos, está formada por uno o más gestores de base de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de acceso a datos.

2.5.1.5- Funcionalidades de Symfony 2.0 utilizadas

➤ Seguridad

La seguridad es un proceso de dos etapas, cuyo objetivo es evitar que un usuario acceda a un recurso al cual no debería tener acceso. En el primer paso del proceso, el sistema de seguridad identifica quién es el usuario obligándolo a presentar algún tipo de identificación. Esto se llama autenticación, y significa que el sistema está tratando de determinar quién eres. Una vez que el sistema sabe quién eres, el siguiente paso es determinar si deberías tener acceso a un determinado recurso. Esta parte del proceso se llama autorización, y significa que el sistema está comprobando si tienes suficientes privilegios para realizar una determinada acción.

➤ Tratamiento de excepciones

Diseñar una buena estrategia de gestión de excepción es importante para la seguridad y la fiabilidad de la aplicación. No hacerlo puede implicar que la aplicación sea vulnerable a ataques de denegación de servicio, o revele información sensible o confidencial. Ante la posibilidad de que ocurra un error inesperado el cual es muy probable que ocurra si de software se trata, la aplicación debe ser capaz de mostrar un mensaje amigable al usuario, evitando revelar cualquier tipo de información asociada a la excepción producida.

La arquitectura base de la aplicación provee un mecanismo que permite tener un control centralizado de las excepciones no tratadas en el sistema. Mediante su uso es posible mapear las excepciones que se produzcan con una vista que informe al usuario del error ocurrido, mediante un mensaje adecuado y permite agregar cualquier lógica asociada a una excepción.

```
class LaboratorioController extends BaseController
{
    public function getGestor() {
        if (!$this->container->has('admon.gtr.laboratorio')) {
            throw new \LogicException('Este servicio no esta registrado en la aplicacion');
        }

        return $this->container->get('admon.gtr.laboratorio');
    }
}
```

Figura 11: Ejemplo de lanzamiento de excepciones

➤ Contenedor de servicios

Un contenedor de servicios o contenedor de inyección de dependencias es un objeto PHP que gestiona la creación de instancias de servicios es decir, de objetos. Este objeto, llamado contenedor de servicios, te permite estandarizar y centralizar la forma en que se construyen los objetos en la aplicación. El contenedor es súper veloz, y enfatiza una arquitectura que promueve el código reutilizable y disociado.

➤ Validación

La validación es una tarea muy común en aplicaciones web. Los datos introducidos en formularios se tienen que validar. Estos también se deben validar antes de escribirlos en una base de datos. Symfony2 viene con un componente Validator que facilita y transparenta esta tarea.

2.5.2- Patrones de diseño

2.5.2.1- Patrones GOF

Los patrones de diseño del grupo de GOF (Gang Of Four) o Pandilla de los Cuatro se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

Decorator (Decorador)

El principal objetivo de este patrón es añadir responsabilidades a objetos concretos de manera dinámica y transparente sin afectar a otros objetos. Este patrón brinda más flexibilidad que la herencia estática y evita que las clases más altas en la jerarquía estén demasiado cargadas de funcionalidad y sean complejas.

En la implementación del sistema este patrón se evidencia en el uso de una plantilla global para las vistas que decorará las demás páginas de la aplicación.

```
{%extends 'AdministracionBundle::admon.html.twig'%}

{%block contenido%

{%include 'ArquitecturaBundle:Grid:GridCrud.html.twig'%}

{%endblock%}
```

Figura 12: Página decorada con la plantilla global

2.5.2.2- Patrones (GRASP)

Los patrones GRASP (Responsibility Assignment Software Patterns) o Patrones generales de software para asignar responsabilidades describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones

➤ Creador

El Creador dirige la asignación de responsabilidades relacionadas con la creación de instancias y su principal objetivo es que el creador de la instancia sea capaz de conectar con la instancia producida en cualquier evento. En la implementación del sistema este patrón se evidencia en las clases Controller¹³ en el momento de crear nuevas instancias de las clases Entidades. Las Controller al recibir de la vista toda la información referente a las entidades son las responsables de crear las mismas y los formularios referentes a dichas entidades.

➤ Experto

El patrón Experto plantea que la clase que debe llevar la responsabilidad es la experta en información, que no es más que aquella clase que contiene toda la información para cumplir con la responsabilidad. Si se le otorga a una sola clase (Experta) la responsabilidad, existe una dependencia nula con otras clases por lo que se garantiza un Bajo acoplamiento y un alto grado de encapsulamiento.

Al igual que el patrón Creador en el desarrollo del sistema este patrón se observa en la creación de las vistas para mostrárselas al usuario, como las clases Controller, que son las expertas en información, son

¹³ Controller: Clase controladora.

las encargadas de crear los formularios, y a través de los mismos se creará la vista que se le mostrará al usuario.

➤ **Bajo acoplamiento**

Cuando una clase depende de muchas otras decimos que existe un alto acoplamiento, esto trae como consecuencia que al realizar un cambio en las clases afines se realizarán cambios locales y que estas clases cuando estén aisladas serán más difíciles de entender, por lo que la mejor práctica es asignar responsabilidades manteniendo un bajo acoplamiento.

Este patrón se observa en el uso del patrón arquitectónico MVC del sistema como las clases de la Vistas no tienen relación con las clases gestoras y las entidades, que representan el modelo, y estas últimas no presentan ninguna relación con los controladores.

➤ **Alta cohesión**

La cohesión es la medida de cuan relacionadas y enfocadas están las responsabilidades de un clase. Una clase con una baja cohesión son difíciles de reutilizar, son delicadas debido a los constantes cambios y son difíciles de comprender por lo que la mejor opción es mantener la complejidad dentro de límites manejables a través de la alta cohesión.

En la implementación del sistema este patrón se ve en la interrelación que existe entre las clases controladora, las clases gestora y los casos de uso. Cada caso de uso presenta una clase controladora y una gestora, la primera encargada de manejar la lógica del negocio del caso de uso y la segunda del manejo de los datos a través de las entidades.

➤ **Controlador**

El objetivo de este patrón es asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades como validaciones, seguridad, entre otras. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.

Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema. Este patrón garantiza un mayor potencial de los componentes reutilizables, garantiza que los eventos del sistema sean manejados por la capa de los objetos del dominio y no por la de la interfaz. Este patrón se evidencia en la solución a través de las clases controladoras que se encuentran

en la carpeta Controller perteneciente a cada Bundle de la solución, y en la estructura de Symfony2 se puede encontrar en el uso del controlador frontal que se encuentra en la carpeta web de cada proyecto de Symfony2.

➤ **Inyección de dependencia**

La inyección de dependencias consiste en pasar a cada componente todo lo que necesita a través de sus constructores, métodos o campos. Este patrón a su vez se deriva de uno más genérico, el patrón de inversión del control. En symfony2 este patrón se evidencia en el trabajo con los servicios y el contenedor de servicios. Un servicio es un objeto PHP que se crea cuando se es necesario utilizar la función a la que hace referencia dicho servicio y un contenedor de servicios (o contenedor de inyección de dependencias) es un objeto PHP que gestiona la creación de las instancias de los servicios. Sin el contenedor de servicios fuera necesario crear cada servicio de forma manual. El contenedor de servicio se encuentra accesible desde cualquier controlador de symfony2 y a través de estos controladores es posible acceder a cualquier servicio mediante el método get() o a través del objeto container.

Las ventajas de utilizar un contenedor de servicios es que un servicio solo es creado cuando se necesita y de igual forma es eliminado una vez que se deja de usar, esto trae un ahorro de memoria y un aumento de velocidad de procesamiento.

2.5.3- Diagrama de clases persistente

A continuación se presenta el diagrama de clases persistentes el cual representa todas las entidades del sistema y sus relaciones.

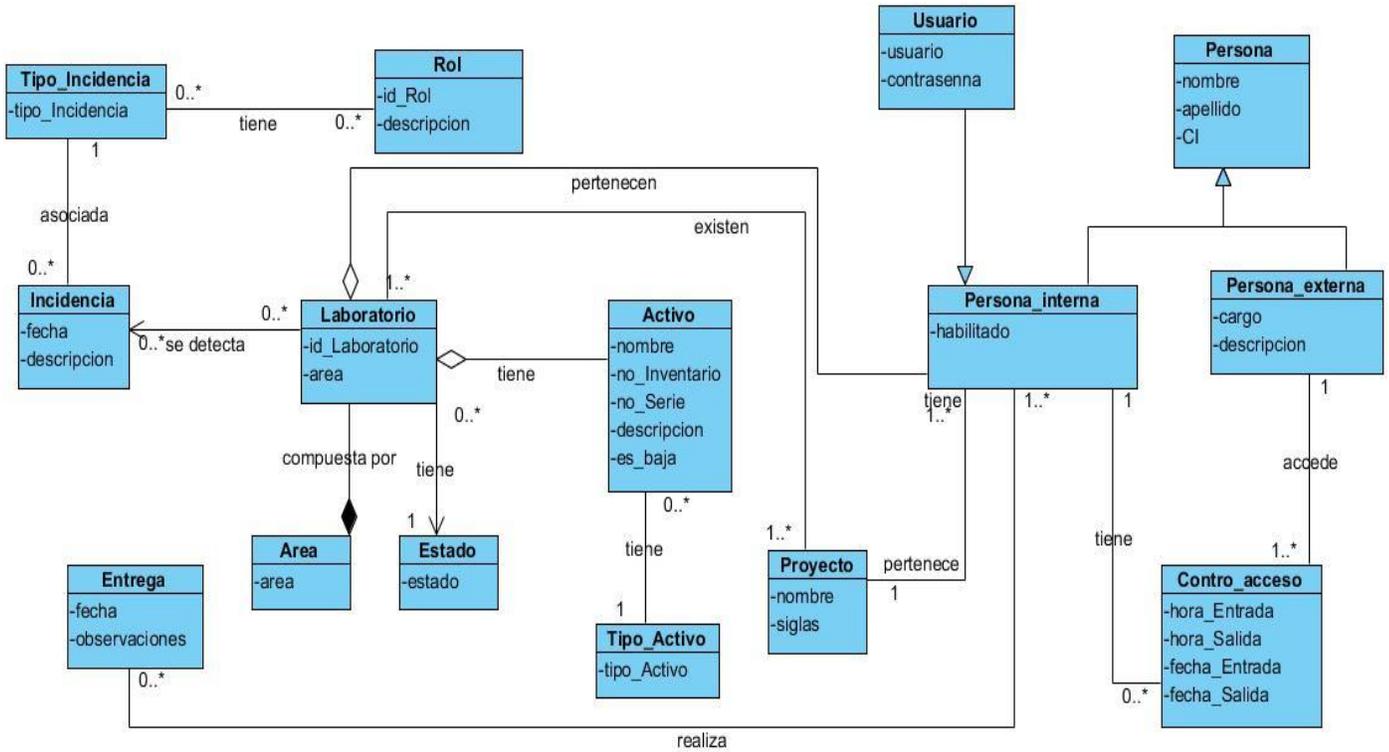


Figura 13: Diagrama de clases persistentes

2.5.4- Diagrama de componente

Los diagramas de componentes modelan la vista estática de un sistema. Se representa como un grafo de componentes de software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten.

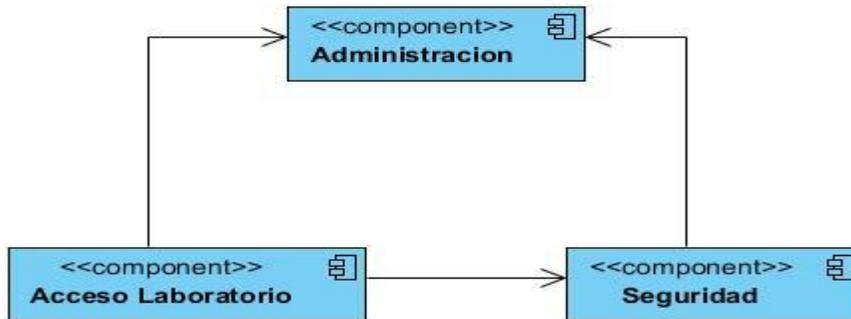


Figura 14: Diagrama de componentes del sistema

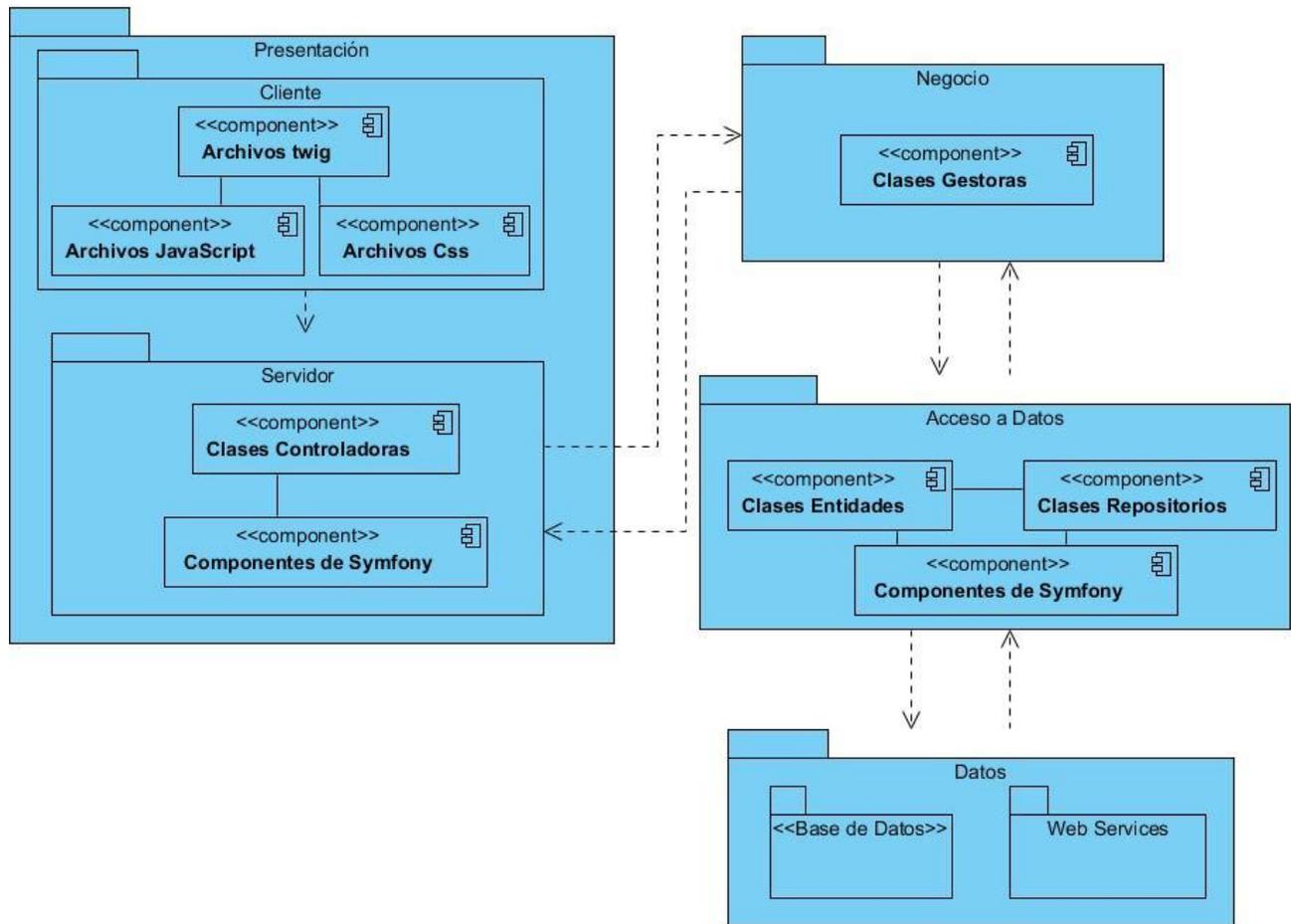


Figura 15: Diagrama de componente basado en la arquitectura del sistema

2.5.5- Diagrama del diseño lógico de los datos

Un diagrama entidad relación permite describir:

- Las estructuras de datos de la base: El tipo de los datos que hay en la base y la forma en que se relacionan.
- Las restricciones de integridad: Un conjunto de condiciones que deben cumplir los datos para reflejar la realidad deseada.
- Operaciones de manipulación de los datos: Son operaciones de agregado, borrado, modificación y recuperación de los datos de la base.

A continuación se muestra el diagrama entidad relación de la aplicación

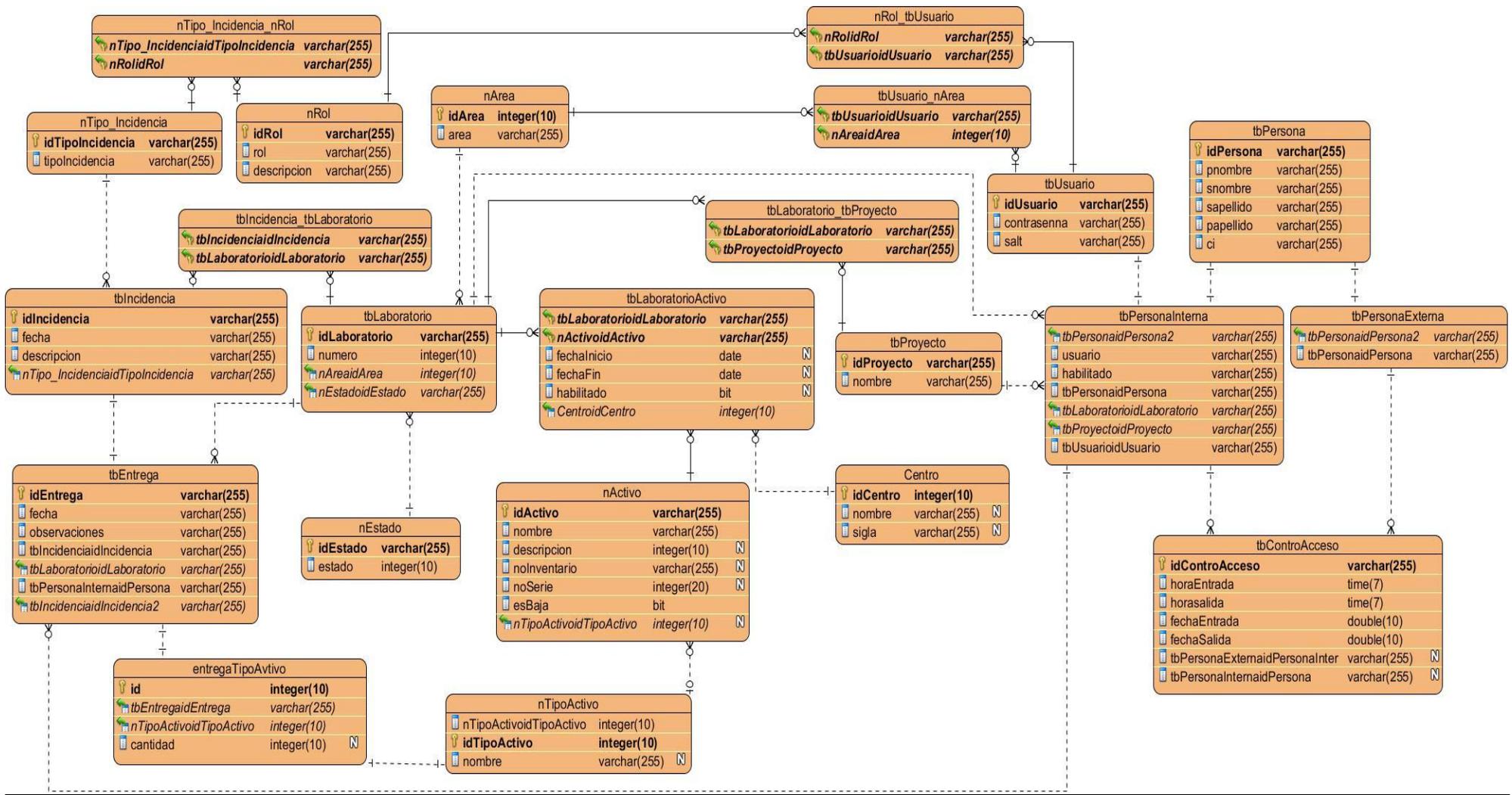


Figura 16: Modelo de datos

2.5.6- Diagrama de clases del Diseño

En un diagrama de clases del diseño se representan cada una de las clases que intervienen en cada uno de los casos de uso del sistema. Se muestra cada una de las vistas o plantillas Twig con sus formularios asociados. También se representan las clases controladoras, las gestoras y las entidades persistentes del sistema y los repositorios. A continuación se presentan un ejemplo de Diagramas de Clases del Diseño para el caso de uso Gestionar Laboratorio:

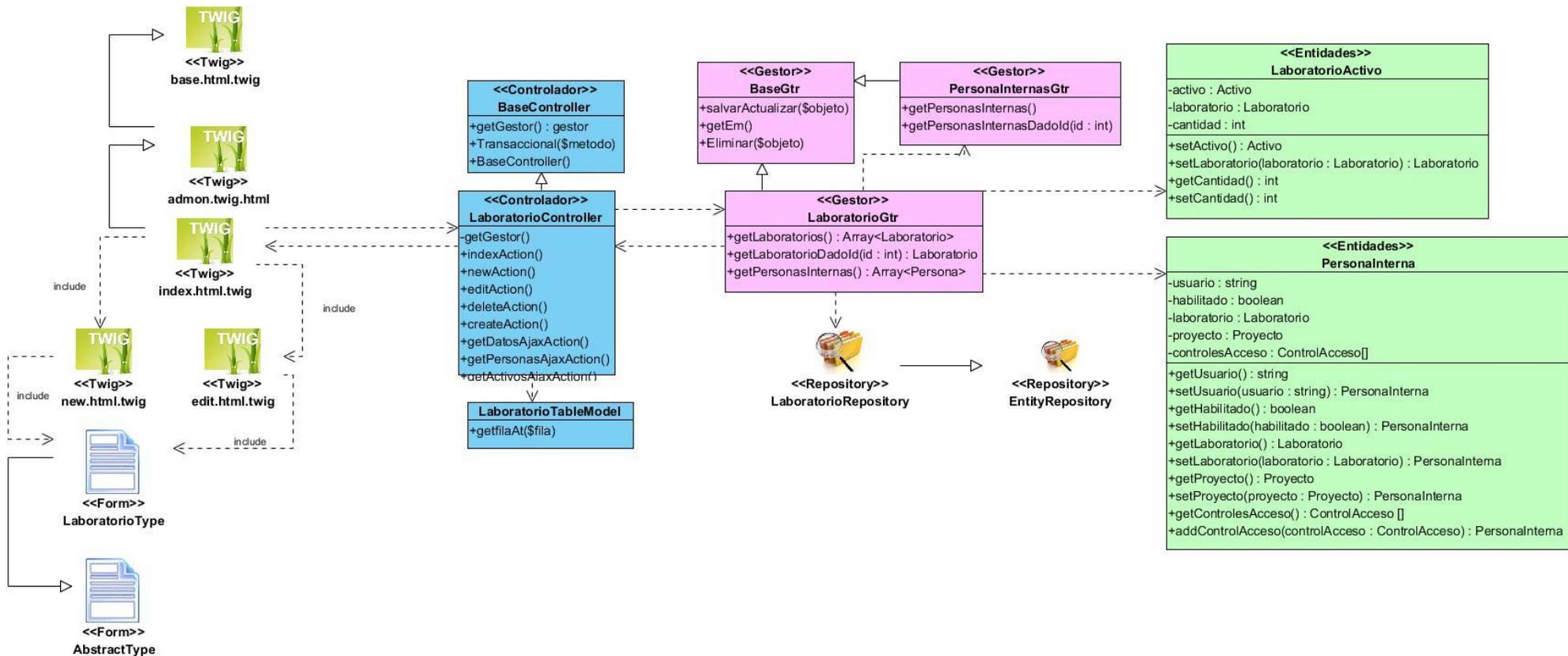


Figura 17: Diagrama de clases del diseño del caso de uso Gestionar Laboratorio

2.6- Implementación

2.6.1- Estándares de codificación

➤ Identación.

El contenido siempre se indentará con tabs, nunca utilizando espacios en blanco.

➤ Cabecera del archivo.

Es importante que todos los archivos .php inicien con una cabecera específica que indique información de la versión, autor de los últimos cambios, entre otros. Es responsabilidad de quien desarrolla decidir si quiere o no agregar más datos.

```
/**
 *
 * @Clase controladora Laboratorio. "LaboratorioController.php"
 * @modificado: 2 de abril del 2013
 * @autor: Mayle
 *
 */
```

➤ Comentarios en las funciones.

Todas las funciones deben tener un comentario, antes de su declaración, explicando que hacen. Ningún programador debería tener que analizar el código de una función para conocer su utilidad. Tanto el nombre como el comentario que acompañe a la función deben bastar para ello.

➤ Ubicación y denominación de archivos.

Se ubicarán los archivos según las convenciones establecidas por Symfony2 o según las especificaciones de la arquitectura utilizada.

Para la denominación de los archivos se seguirán las convenciones establecidas por Symfony2.

Ejemplo:

Para las clases de gestión del negocio se usará el sufijo Gtr:

LaboratorioGtr

Para los tablemodel definidos para los grid se usará el sufijo TableModel:

LaboratorioTableModel

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

Para las entidades de presentación se usará el sufijo EP:

LaboratorioActivoEP

En las páginas, plantillas html.twig definidas para la vista el nombre del archivo debe seguir el estándar de la denominación de las clases. Ejemplo:

index.html.twig

En caso de estar formado por más de una palabra:

realizarEntrega.html.twig

Para la denominación de los servicios a crear se tendrán en cuenta los siguientes elementos. Para los parámetros se utilizará la siguiente nomenclatura (todo en minúscula utilizando como separador el carácter punto (.))

bundle.categoria.class:

Ejemplo:

admon.gtr.laboratorio.class
admon.tm.laboratorio.class:

Nota: las categorías más comunes son “gtr” y “tm”.

Para los servicios se utilizará la siguiente nomenclatura (todo en minúscula utilizando como separador el carácter punto (.)). Es la misma nomenclatura que el parámetro pero sin el sufijo “.class”.

bundle.categoria:

Ejemplo:

admon.gtr.laboratorio:

Para las rutas se utilizará la siguiente nomenclatura (todo en minúscula utilizando como separador el carácter guión bajo (_))

bundle_controlador_action:

Ejemplo:

```
admon_laboratorio_new:  
  pattern: /admon/laboratorio/new  
  defaults: {controller: AdministracionBundle:Laboratorio:new }
```

El patrón de la ruta debe tener relación con la acción que se llama y ser lo más humano posible ya que es lo que el usuario percibe desde el navegador.

Para el caso del patrón donde el nombre esté compuesto por más de 1 palabra serán separadas por "-".

Ejemplo:

```
pattern: /admon/laboratorio-activo/new
```

Nota: Los nombres de las rutas y servicios deben de iniciar con letra minúscula. Si el nombre de una ruta o servicio se comprende de más de una palabra, la primera letra de cada nueva palabra debe capitalizarse.

➤ Clases

Las clases serán colocadas en un archivo .php aparte, donde sólo se colocará el código de la clase. El nombre del archivo será el mismo de la clase. Las clases siguen las mismas reglas de las funciones, por tanto, debe colocarse un comentario antes de la declaración de la clase explicando su utilidad. Los nombres de las clases deben de iniciar con letra mayúscula, en caso de que sea de más de una palabra, la primera letra de cada nueva palabra debe capitalizarse. No se permiten las letras capitalizadas sucesivas; por ejemplo, una clase "SymfonyPDF" no se permite, mientras "SymfonyPdf" es aceptable.

Siempre utilizar las etiquetas `<?php ?>` para abrir un bloque de código. No utilizar el método de etiquetas cortas.

2.6.2- Diagrama de despliegue

El modelo de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los links de comunicación entre ellos, y las instancias de los componentes y objetos que residen en ellos. El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento, y las conexiones entre estos elementos en el sistema. A continuación se muestra el diagrama de despliegue correspondiente a la aplicación.

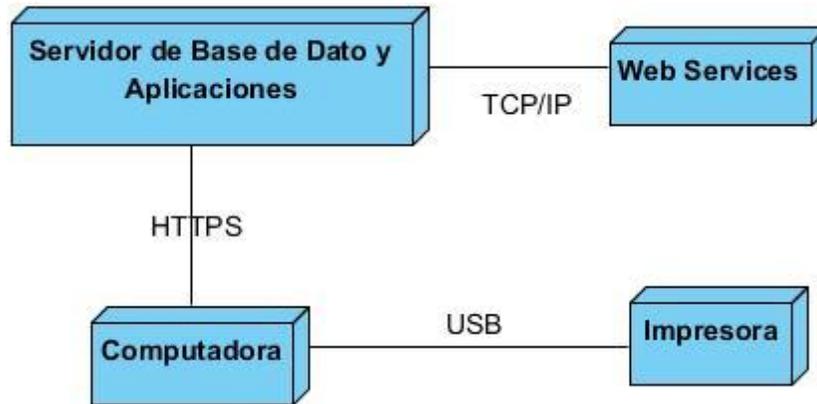


Figura 18: Diagrama de despliegue

2.7- Conclusiones parciales

En este capítulo se abordó lo referente al modelado del negocio, lo relacionado con el análisis y diseño de la propuesta de solución permitiendo un mejor entendimiento para la implementación del sistema. Se describió la arquitectura y los patrones utilizados además de realizar los diagramas de caso de uso, modelo de datos y diagramas de clases del diseño que permiten un mejor entendimiento para el programador. También se analizaron temas relacionados con la implementación, describiéndose así los estándares de codificación empleados en la propuesta de solución, se confeccionó el diagrama de componente y despliegue, todo esto permitió obtener una descripción sobre los procesos de implementación e implantación del producto desarrollado.

CAPÍTULO 3: VALIDACIÓN

3.1- Introducción

La aparición de errores en el desarrollo de software tiene una alta probabilidad, pues el factor humano aún está lejos de lograr la perfección, por lo que de alguna manera debe asegurarse la calidad del producto, y aquí es precisamente donde entran a jugar su papel las pruebas y validaciones al software, estas son un elemento esencial para el aseguramiento de la calidad del mismo. Estos errores pueden estar dados por la mala especificación de los requisitos, uso indebido de las estructuras de datos, entre otras causas.

Este capítulo trata acerca de la validación de la solución propuesta anteriormente, a través de pruebas de software que garanticen el correcto funcionamiento y totalidad de las funcionalidades especificadas por el cliente. También se abordará temas relacionados con probar la necesidad que tiene hoy CEGEL de tener un control del personal que accede a sus laboratorios de producción, y de sus activos. Para ello se va a utilizar la técnica del panel de experto para determinar el por ciento de cumplimiento de algunos identificadores, se confeccionará un caso de estudio al cual se le va a aplicar el funcionamiento del sistema y con los resultados obtenidos por ambos métodos de evaluación se podrá hacer una comparación de cuanto mejora el sistema desarrollado, la gestión de la seguridad informática en los laboratorios del centro.

3.2- Validación del modelado de negocio

La validación de los artefactos generados como parte de la modelación del proceso de negocio se realizó aplicando la técnica Revisión técnica formal, donde los clientes revisaron cada documento generado con el objetivo de validar su correcta elaboración y que el flujo de actividades de cada proceso estuviera en correspondencia con la información brindada. Como constancia de esta revisión se emite un Acta de aceptación (consultar documento entregable Acta de aceptación_1).

3.3- Validación de requisitos

La validación de los requisitos se realiza con la finalidad de comprobar que los requisitos identificados sean precisos, consistentes, reales, verificables, definan lo que el usuario desea del producto final, que los errores detectados sean corregidos y el resultado del trabajo cumpla con los estándares establecidos para el proceso (41).

Para la validación de los requisitos fueron aplicadas las siguientes métricas:

Estabilidad de los requisitos

El objetivo de esta métrica es medir cuan estables son los requisitos para asegurar su adecuación antes de pasar a la próxima disciplina. Se considera que los requisitos son estables cuando no existen adiciones o supresiones en ellos que impliquen modificaciones en las funcionalidades principales de la aplicación. La estabilidad de los requisitos se calcula como:

$$ETR = \left[\frac{RT - RM}{RT} \right] * 100$$

Donde:

ETR: Estabilidad de los requisitos.

RT: Total de requisitos definidos [54].

RM: Cantidad de requisitos modificados [4].

Luego de aplicada esta métrica a los requisitos identificados de la aplicación arrojo como resultado que la estabilidad de los mismos es de un 92% lo que demuestra que no se están realizando cambios sobre los requisitos, que son estables y, por tanto, es confiable trabajar el análisis y diseño sobre ellos.

Especificidad de los requisitos

El objetivo de esta métrica es cuantificar la especificidad o falta de ambigüedad en la definición de los requisitos. Para calcular esta métrica deben contarse los requisitos que tuvieron igual interpretación por los revisores y compararlos con el total de requisitos definidos. Dicha especificidad se calcula como:

$$Q_1 = \frac{n_{ui}}{n_r}$$

Donde:

Q1: Especificidad de los requisitos.

Nr: Total de requisitos definidos [54].

Nui: Total de requisitos para los que los revisores tuvieron interpretaciones idénticas [50].

El cálculo de esta métrica arrojó como resultado 0.92, lo que indica que mientras más cercano a 1 se encuentre este valor mayor será la consistencia de la interpretación de los revisores para cada requisito y menor será la ambigüedad en la especificación de los mismos.

Grado de validación de los requisitos

Los requisitos deben ser posibles de validar. Esta validación se realiza en consenso del equipo de desarrollo al contrastar lo que desea el cliente con la posibilidad real de implementarlo. El grado de validación de los requisitos mide la corrección en la definición de los mismos. Este valor se calcula como:

$$Q_3 = \frac{n_c}{(n_c + n_{nv})}$$

Donde:

Q3: Grado de validación de requisitos.

Nc: Total de requisitos validados correctamente [50].

Nnv: Total de requisitos no validados [4].

El valor obtenido luego de aplicar esta métrica es 0.92 lo cual indica un alto nivel de corrección en la definición de los requisitos.

3.4- Validaciones del diseño

Una vez realizado el diseño del sistema de Gestión de los laboratorios de CEGEL, se dio paso a su evaluación, para esto se utilizaron las métricas orientadas a objetos, específicamente a las clases, ya que las medidas y métricas para una clase individual, la jerarquía y las colaboraciones de estas, permiten al ingeniero de software evaluar la calidad del diseño propuesto.

Se empleó la métrica **Tamaño de clase (TC)**, para aplicarla se tuvo en cuenta la cantidad de procedimientos que tenían las clases, luego se realizó el cálculo del promedio de los procedimientos y mediante un criterio se obtuvo la categoría (pequeño, mediano, grande), para la Responsabilidad, Complejidad y Reutilización. Debido a que el sistema cuenta con gran cantidad de clases para esta métrica solo se tuvo en cuenta las clases Controller y las Gestores.

Tabla VI: Formas de cálculo para el TC

Clasificación	Valores de los umbrales
Pequeño	\leq Promedio de operaciones(PO)
Medio	$>$ PO y $\leq 2*PO$
Grande	$> 2*PO$

En este momento se procede a la definición del tamaño de las clases según sus atributos y operaciones de cada una de ellas. Ver Anexo #2. Luego de aplicada la métrica a un conjunto de clases de la aplicación, la misma arrojó como resultado el siguiente:

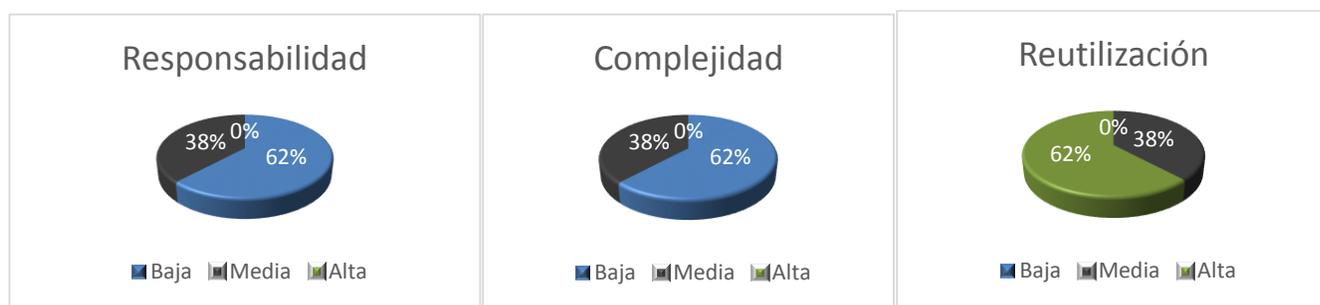


Figura 19: Representación de la incidencia de los resultados de la evaluación de la métrica TC

Luego de realizar un análisis de los resultados obtenidos en el empleo de esta técnica, se concluye que el diseño realizado es simple y tiene una calidad aceptable, pues la mayoría de las clases que fueron analizadas se encuentran dentro de la categoría de pequeñas, demostrándose en los pequeños valores de TC alcanzados que la implementación de forma general es sencilla, se disminuye en gran medida la responsabilidad de las clases lo que significa que las clases existentes dentro del sistema se puedan reutilizar ampliamente.

Relaciones entre Clases (RC)

Esta métrica está basada en el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de Calidad de acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas. En la Tabla VII se muestra las relaciones de uso de las clases seleccionadas.

Tabla VII: Relaciones de uso entre clases

Clase	Cantidad de Relaciones de Uso
EntregaController	5
CentroController	5
Activo	1
Laboratorio	1
ActivoRepository	3
LaboratorioRepository	2
PersonalInternaRepositoryr	3
TipoIncidenciaRepository	3
Centro	1
Persona	1

Resultado de la aplicación de la métrica RC

La métrica de diseño RC fue aplicado a una muestra de 10 clases seleccionadas aleatoriamente. Las figuras 20 y 21 muestran los resultados que arrojó la aplicación de la métrica en los atributos de calidad acoplamiento, complejidad de mantenimiento, reutilización y cantidad de pruebas.

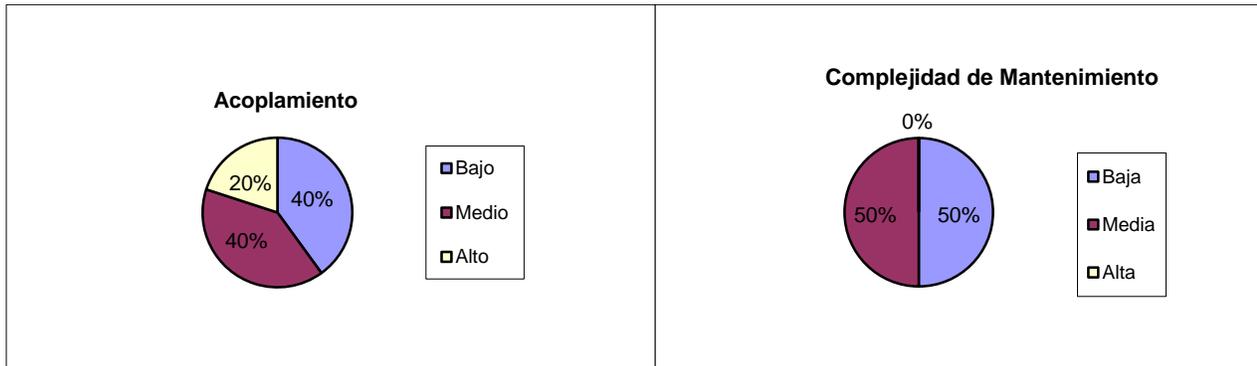


Figura 20: Resultado de la métrica RC al atributo Acoplamiento y Complejidad de mantenimiento

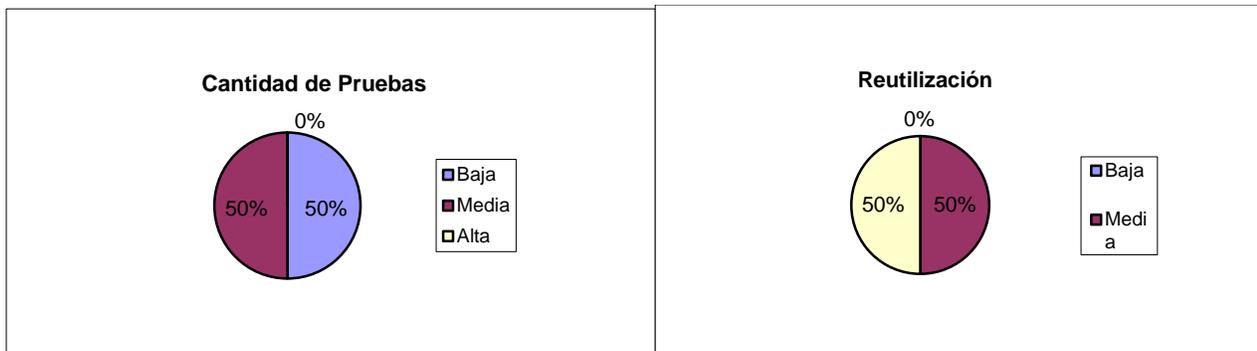


Figura 21: Resultado de la métrica RC al atributo Acoplamiento y Complejidad de mantenimiento

3.5- Pruebas de software

De acuerdo a la IEEE el concepto de prueba se define como: “Una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente”. De aquí la importancia que tiene realizarle pruebas al software antes de ser entregado al usuario final, garantizando así que el mismo tenga la menor cantidad de errores y mejor calidad posible.

3.5.1- Aplicación de las pruebas de caja negra

Entre las técnicas de caja negra que existen, se utilizó la prueba de partición de equivalencia. Esta técnica divide el campo de entrada de un programa en clases de datos de las cuales se puede obtener los casos de pruebas correspondientes. Mediante la aplicación de esta se examinó los valores válidos e inválidos de las entradas existentes en el software, descubriendo de forma inmediata clase de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

Para definir las clases de equivalencia se tuvieron en cuenta un conjunto de reglas:

- Si una condición de entrada especifica un rango, entonces se confeccionan una clase de equivalencia válida y dos inválidas.
- Si una condición de entrada especifica a cantidad de valores, se identifica una clase de equivalencia válida y dos inválidas.
- Si una condición de entrada especifica un conjunto de valores de entrada y existen razones para creer que el programa trata en forma diferente a cada uno de ellos, se identifica una clase válida para cada uno de ellos y una clase inválida.
- Si una condición de entrada especifica una situación de tipo “debe ser”, se identifica una clase válida y una inválida.

Luego de tener las clases válidas e inválidas definidas, se procedió a definir los casos de pruebas, para todos los casos de uso. Cada uno de estos casos de pruebas se definió teniendo en cuenta lo siguiente:

- Escribir un nuevo caso que cubra tantas clases de equivalencia válidas no cubiertas como sea posible hasta que todas las clases de equivalencia hayan sido cubiertas por casos de prueba.
- Escribir un nuevo caso de prueba que cubra una y solo una clase de equivalencia inválida hasta que todas las clases de equivalencias inválidas hayan sido cubiertas por casos de pruebas.

Para ver los datos casos de prueba correspondiente al caso de uso Gestionar Laboratorio, Ver Anexo # 3. Los restantes casos de pruebas se encuentran anexados a los artefactos generados de la presente investigación. Durante el proceso de la confección de los casos de pruebas en una primera iteración se detectaron un total de 22 no conformidades entre errores ortográficos, de validación y funcionales. En una segunda iteración se detectaron un total de 9 no conformidades de las cuales todas fueron satisfactoriamente corregidas para lograr la puesta en práctica de la aplicación por el centro CEGEL.

3.6- Análisis cuantitativo de la solución

3.6.1- Técnica del panel de expertos

Para la utilización de la técnica del panel de experto primero se procedió a elegir cada uno de los expertos para ellos se tuvo en cuenta que fueran personas que pertenecieran a cada uno de los laboratorios para tener los datos más reales posibles. Estos datos se recogen en una tabla donde se presentan cada uno de los expertos y el laboratorio que le corresponde como escenario de muestras. Ver Anexo # 4.

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

Se seleccionó un total de 5 expertos, los cuales se reunieron para determinar los principales indicadores a medir. Se realizó una tormenta de idea para ver los posibles indicadores de ellos los que tuvieron una mayor aceptación por parte del grupo de experto fueron los seleccionados para realizar el análisis. Estos son:

1. Cantidad de entradas registradas a un laboratorio y cantidad real de personas que accedieron sin registrarse
2. Cantidad de salidas registradas a un laboratorio y cantidad real de personas que salieron sin registrarse.
3. Cantidad de entradas de personal externo registradas y cantidad real de personas externas que accedieron sin registrarse.
4. Cantidad de salidas de personal externo registradas y cantidad real de personas externas que salieron sin registrarse.
5. Cantidad de activos registrada en el inventario por laboratorio y cantidad real existente en los mismos.
6. Cantidad de movimientos de activos registrada y cantidad real de movimientos existentes.
7. Cantidad de incidencias registradas y cantidad real de incidencias existentes.

De cada uno de estos indicadores se calcula el por ciento de cumplimiento. La fórmula para dicho cálculo es la siguiente:

Porcentaje de cumplimiento (% Cump) = Total Registrado (TReg) * 100/ Total Real (TR)

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

Tabla VIII: Evaluación de los expertos por cada indicador

Ex	Ind 1	% Cump	Ind 2	% Cump	Ind 3	% Cump	Ind 4	% Cump	Ind 5	% Cump	Ind 6	% Cump	Ind 7	% Cump
1	0/50	0	0/50	0	0/5	0	0/5	0	90/94	96	2/4	50	0/3	0
2	0/40	0	0/40	0	0/7	0	0/7	0	96/96	100	0/1	0	0/1	0
3	0/42	0	0/42	0	0/3	0	0/3	0	98/112	87	0/0	0	0/2	0
4	0/35	0	0/35	0	0/1	0	0/1	0	99/100	99	0/2	0	1/1	100
5	0/37	0	0/37	0	0/2	0	0/2	0	95/99	95	1/3	33	1/4	25
% Cump		0		0		0		0		95		17		25

3.6.2- Caso de estudio

En los laboratorios de producción del Centro de Gobierno Electrónico acceden diariamente gran cantidad de personas entre estudiantes, profesores, trabajadores, y personal externo. De estas personas se registra su horario de entrada y salida en cada una de las secciones mañana, tarde y noche en caso que visiten el laboratorio. En estos lugares se encuentran ubicados la mayor cantidad de activos que tiene el centro. De los mismos se controla su número de serie o número de inventario y su ubicación física, es decir la cantidad de activos por cada laboratorio. Por problemas ajenos al centro en muchas ocasiones es necesario realizar movimientos con los activos, esto lleva que se actualice el inventario de los laboratorios de los cuales se realiza el cambio y con ello se logra tener un estado real de donde se encuentra cada activo en cualquier momento. En estos laboratorios también se incurren en problemas tanto de mal uso de los medios informáticos por parte del personal que acceden a ellos como pérdida de activos, estas incidencias se registran y notifican a la persona encargada de su solución.

Solución arrojada por el sistema

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

Luego de implantarse el sistema con los indicadores que se describen en el caso de estudio se tuvieron los siguientes resultados:

Tabla IV: Resultado arrojado por el Sistema con relación al acceso de las personas

Total de personas	Total de Personas Internas (PI)	Total de Personas Externas (PE)	Ind 1 % Cump	Ind 2 % Cump	Ind 3 % Cump	Ind 4 % Cump
40	35	5	100	100	100	100

Tabla X: Resultado arrojado por el Sistema con relación al control de activo

Total de activos	Total de Activos registrado en inventario	Ind 5 % Cump	Ind 6 % Cump
112	112	100	100

Tabla XI: Resultado arrojado por el Sistema con relación al control de incidencias

Total de incidencias realizadas	Ind 7 % Cump
18	100

3.6.3- Valoración crítica

Luego de realizarse la técnica del panel de experto la cual arrojó como resultados que el cumplimiento de los indicadores de primera necesidad para centro CEGEL es muy malo, debido a que no se realizan los registros de las entradas y salidas tanto del personal interno como externo, provocando así que no se conozca la ubicación de las personas, ni si cumple o no su jornada laboral, no se conoce el motivo ni la causa por la cual alguien externo al laboratorio accede. El inventario de los laboratorios se encuentra desactualizado, provocando así que exista pérdida de equipos y no se sepa quién es el responsable, cuando se realiza el movimiento de un activo solo en el 17% de las veces se registra o notifica y de las incidencias que se suceden en estos lugares son informadas un 25%. De esta forma queda demostrado que solo el 19% de los indicadores a medir es el que se cumple el resto es violado por cada una de las personas

encargadas de su cumplimiento. Por todo lo antes expuesto es que surge la necesidad de crear un sistema para la Gestión de los laboratorios en el centro.

A este sistema se le realiza una prueba mediante la elaboración de un caso de estudio, puesto que no hay el suficiente tiempo para su puesta en práctica, el mismo arrojó como resultados que se cumplen en un 100% los indicadores que hoy son de primera necesidad para el centro y con ello queda probado la mejora de la gestión de la seguridad en estos lugares.

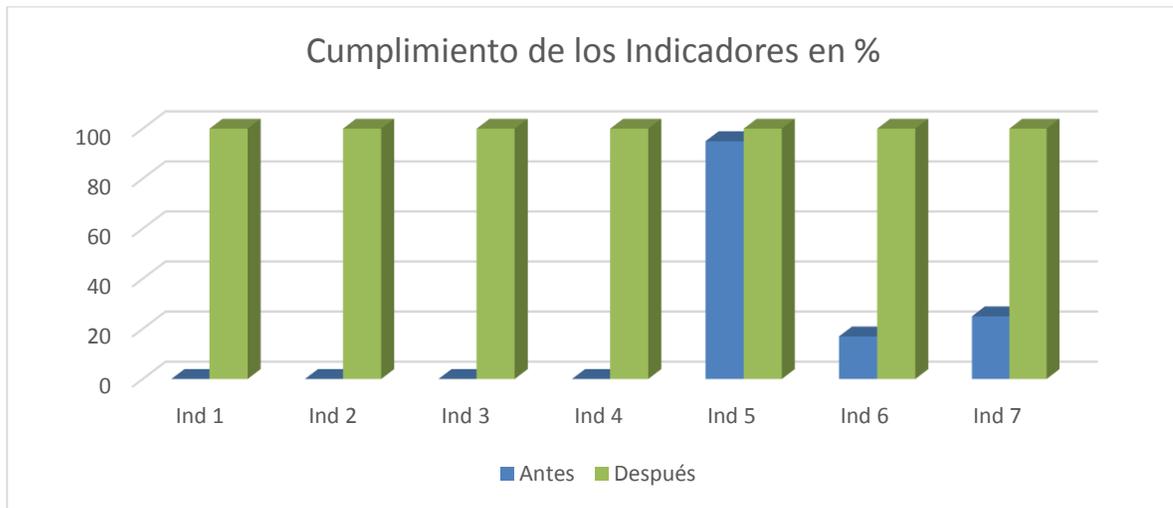


Figura 22: Cumplimiento de los indicadores en % antes y después del sistema

3.7- Conclusiones parciales

Con el desarrollo de este capítulo se pudo apreciar la importancia de la realización de las pruebas en el proceso de desarrollo de software. Se realizaron las validaciones correspondientes en las etapas de modelado del negocio, análisis y diseño de la propuesta de solución permitiendo que no existieran deficiencias a la hora de implementar el sistema. Se desarrollaron las pruebas de caja negra, las cuales de manera general arrojaron resultados satisfactorios, siendo corregidos todos los errores encontrados por las mismas. Se realizó un análisis cuantitativo de la propuesta de solución el cual permitió validar que la solución propuesta cumple con el problema que tenía el centro.

CONCLUSIONES

En el presente trabajo se dio respuesta al problema planteado referente a la necesidad de desarrollar una aplicación para la Gestión de los laboratorios de CEGEL, y se dio cumplimiento al objetivo general de esta investigación. Se sentaron las bases tecnológicas para el desarrollo y puesta en práctica del sistema permitiendo tener un control estricto del personal que accede a estos lugares como de los activos que allí se tienen.

Asimismo se cumplieron los objetivos específicos trazados, se elaboró el marco teórico de la investigación que permitió conocer en detalle lo que se quería desarrollar y con ello tener una correcta selección de las herramientas tecnológías y lenguajes para la elaboración del sistema. También se desarrolló la propuesta de la solución la cual permitió dar respuesta a la problemática planteada y se validó la solución para así demostrar el cumplimiento de los objetivos planteados en la investigación.

RECOMENDACIONES

Las siguientes recomendaciones se deben tener en cuenta para el posterior desarrollo y mejora del sistema:

- Integrarlo con GLPI para lograr tener una gestión completa de las características de los activos.
- Implementar otros procesos relacionados con la gestión de laboratorios que no han sido incluidos en la propuesta de solución.

REFERENCIAS BIBLIOGRÁFICAS

1. CEGEL - Centro de Gobierno Electrónico; <http://gespro.cegel.prod.uci.cu/>; Visitado el 29 de Mayo de 2013.
2. Tecnología de la Información; Código de buenas prácticas para la gestión de la seguridad de la información; NC ISO/IEC FDIS 17799:2007.
3. Estudio de la contabilidad general; Maldonado; Ediciones MES; ISBN: 980-6034-78-3.
4. Pfleeger, Charles P. Security in computing. 2006. ISBN: 978-0-13-239077-4.
5. Principios básicos de Seguridad de la Información; ISO 17799:2000 y BS 7799-2:2000.
6. Gómez, Vieites and Álvaro; Enciclopedia de la Seguridad Informática; 2007; ISBN 9788478977314.
7. Sánchez, Jordi; Computación Distribuida: Sistema de Autenticación y Autorización en Internet; MIPO Master en Interacción Persona Ordenador; Universidad de Lleida; Junio de 2011; Visitado el 5 de Octubre de 2012.
8. Real Academia Española; <http://www.rae.es/RAE/Noticias.nsf/Home?ReadForm>; Visitado el 15 de Octubre de 2012.
9. Sistema de Información para la Gestión de Activos SIGA; <http://siga.centraldeinversiones.com.co/siga/>; 2009; Visitado el 1 de Noviembre de 2012.
10. IBM; <http://www-01.ibm.com>; Visitado el 3 de Noviembre de 2012.
11. Tecnología Pyme; <http://www.tecnologiapyme.com/software/gipi-completa-aplicacion-de-codigo-abierto-para-gestionar-el-inventario>; Visitado el 5 de Noviembre de 2012.
12. Xperta; <http://www.xperta.es/>; Visitado el 11 de noviembre de 2012.
13. Berzosa, Luis Rodríguez; Control de accesos: De la era mainframe a las PKI; <http://www.iec.csic.es/>; 24 de enero de 2012; Visitado el 6 de Octubre de 2012.
14. Preguntas sobre códigos de barras; <http://www.barmax.com/>; 1999; Visitado el 8 de Octubre de 2012.
15. Luna Téllez, Linda; Metodología Ágil Vs Metodología Tradicional; Modulo de calidad; Universidad Gabriel Rene Moreno; 2012.
16. METODOLOGIAS AGILES; Proceso unificado Ágil (AUP); Universidad Unión Bolivariana; 2013.
17. Modelado de Procesos; Teoría de sistemas; 1er Semestre; Universidad Valparaíso; 2005.
18. Madrid, Víctor Javier; Introducción a BPMN; Tutorial; 07-06-2010.

APLICACIÓN WEB PARA LA GESTIÓN Y ACCESO

A LOS LABORATORIOS DE CEGEL

19. Ruiz, F.; Aplicación de Métricas Software en la evaluación de Modelos de Procesos de Negocio; México-Ciudad Real, España.
20. Eguiluz, Javier; Desarrollo web ágil con Symfony2; 15-10-2012.
21. Zend Framework; Manual en Español; 14 de Noviembre del 2009.
22. EXT JS; <http://www.sencha.com/>; 2013; Visitado el 14 de marzo de 2013.
23. Alvarez, Miquel Angel; Manual de jQuery; 2013.
24. Practical symfony, second edition; symfony 1.2 | Propel; 2010.
25. Doctrine ORM for PHP; Guide to Doctrine for PHP; 2010.
26. Twitter Front-end Summit; Página 5-31; 2011.
27. Rumsbaugh, James and Jacobson, Ivar El.al Booch; El lenguaje unificado de modelado; ISBN: 84r7829r037r0.
28. Heredia and Cristian Van El.al Herminio; Introducción al PHP; 23 de Mayo de 2001.
29. Gutiérrez, David; Desarrollo web y tecnología; 28 de Julio de 2011.
30. Introducción a JavaScript; <http://www.librosweb.es/>; 2012; Visitado el 16 de Octubre de 2012.
31. Html Castillo Cantón, Alejandro; Manual de HTML5 en español; 2011.
32. Css Alvarez, Miquel Angel; Manual de Css 3; 2013.
33. María A. Mendoza Sanchez. Metodologías de desarrollo de software, 2004.
34. Larman, C; UML y Patrones. Introducción al análisis y diseño orientado a objetos. 1999, México.
35. Eclipse Com; <http://www.eclipse.org/>; 2013; Visitado el 24 de Octubre de 2012.
36. NetBeans; <http://netbeans.org/features/index.html>.; Visitado el 3 de mayo de 2013.
37. Herrera, Érica and Uribe; Servidor de aplicaciones; 2009.
38. MySQL 5.0 Reference Manual; Página 4 -29; 2011.
39. Badillo, Camilo and Mahecha, Esteban; Mapas de proceso; 2012.
40. Somerville, Ian; Ingeniería de Software; Séptima edición; 2005.
41. Pressman, Roger S; Ingeniería del Software Un enfoque práctico; Sexta edición; 2005.