

Universidad de las Ciencias Informáticas

Facultad 3



**“Desarrollo del componente Seguridad y Salud en el Trabajo
del subsistema Capital Humano del Sistema Integral de
Gestión Cedrux”**

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

Autores: Lisbet Cedeño Ávila
Hazel Cabrera Ortega

Tutor: Ing. Dailys Díaz Fuentes.

Co-tutor: Ing. Armando Lázaro Simón Quintero.

La Habana, Junio 2013.
“Año 55 de la Revolución”

DECLARACIÓN DE AUTORÍA

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

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

Hazel Cabrera Ortega

Autor

Lisbet Cedeño Ávila

Autor

Ing. Dailys Díaz Fuentes

Tutor

Ing. Armando Lázaro Simón

Co-tutor

DATOS DE CONTACTO

Datos del tutor:

Ing. Dailys Díaz Fuentes

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: dfuentes@uci.cu

Datos del co-tutor:

Ing. Armando Lázaro Simón Quintero

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: alsquintero@uci.cu

DEDICATORIA

Lisbet:

Dedico este trabajo a mis padres, por ser una guía incondicional durante mi vida, por su comprensión, paciencia, respeto, por el amor y apoyo absoluto que me han brindado en todo momento, los admiro y estoy orgullosa de seguir sus pasos...

A mi novio por si mi sostén en los momentos de flaqueza, por su constante preocupación, por su inmenso cariño, amistad y respeto, te quiero...

A mi familia que ha sido la base de mi crianza, gracias por ser unidos, preocupados y cariñosos...

A la familia pinareña que he ganado durante estos cinco años, que me han entregado un inmenso cariño, confianza y me han acogido como a una hija, mil gracias a todos...

Hazel:

Esta tesis quiero dedicarla de forma muy especial a mis padres, por ser la razón de que haya llegado hasta aquí, por brindarme su apoyo en cada momento, por escucharme siempre, por su confianza, por haberme hecho el hombre que soy.

A mi madrastra Cristina por su dedicación y apoyo, a la cual considero es mi segunda madre.

A mi padrastro Yosleyvys que a pesar del poco tiempo que lleva a mi lado se ha comportado conmigo con un padre con su hijo, el cual llego para darle estabilidad a mi familia.

A mis hermanas, para que sepan que cuando se quiere se puede y para que les sirva de ejemplo y se hagan mujeres de bien

A mis suegros Clara y Omar por toda su ayuda y su apoyo, a los cuales considero son mis terceros padres.

A mi novia Yaneisi por compartir conmigo todos estos años y darme los momentos más felices que viví aquí y por sobre todas las cosas por enseñarme a ser la persona que siempre quise ser.

AGRADECIMIENTOS

Queremos agradecerles a nuestros padres que representan lo más valioso en nuestras vidas y a toda nuestra familia por su comprensión y ayuda en todo momento.

A nuestros tutores y a toda la gente del proyecto, especialmente Aylin, Yarenis y Arnolis que pusieron su granito de arena para la realización de este trabajo.

A todas nuestras amistades por los buenos momentos que pasamos y los recuerdos que hoy llevamos con nosotros, en especial a los más cercanos que han compartido con nosotros momentos de alegría, tristeza, estrés, ocio y nos han llegado a conocer un poquito mejor.

Lisbet:

A mi novio Carlos por creer en mí y brindarme su apoyo incondicional durante los cinco años de carrera.

A mi compañero de tesis por ser paciente, dedicado y trabajar sin descanso hasta lograr nuestro objetivo.

Hazel:

A mi novia Yaneisi por todos los momentos que estuvo a mi lado, por su apoyo durante todo este tiempo en la universidad.

A mi compañera de tesis, por el buen trabajo en conjunto que supimos desarrollar, lo cual conllevó a obtener excelentes resultados en el presente trabajo y por consiguiente el surgimiento de una buena amistad.

Muchas gracias a todos!!!

RESUMEN

Los procesos de gestión de Seguridad y Salud en el Trabajo en las entidades presupuestadas y empresariales cubanas no engloban una solución integral. Esto provoca demora en los servicios, pérdida de la información y que los reportes estadísticos no sean los más confiables en las entidades donde la gestión de este proceso se lleva a cabo de forma manual. Para darle solución al problema anterior, se propone el presente trabajo teniendo como objetivo fundamental desarrollar el componente Seguridad y Salud en el Trabajo del subsistema Capital Humano del sistema integral de gestión Cedrux. Para el desarrollo del componente se utilizó como lenguaje de programación del lado del servidor PHP en su versión 5.2.6 y del lado del cliente Java Script aplicándose ambos mediante el IDE Netbeans en su versión 7.0.1. La especificación, construcción y documentación de la solución se realizó basándose en el Modelo de Desarrollo del centro CEIGE¹. Como resultado se obtuvo una solución capaz de gestionar la información de forma correcta que a diferencia de otros sistemas permite una gestión integral con funcionalidades generales de los procesos de Seguridad y Salud en el Trabajo y las particularidades de la economía cubana.

Palabras claves: Capital Humano, Cedrux, entidades presupuestadas y empresariales, seguridad y salud en el trabajo, Componente.

¹CEIGE: Centro de Informatización de la Gestión de Entidades.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN..... 1

1. FUNDAMENTACIÓN TEÓRICA..... 4

 1.1 Introducción 4

 1.2 Principales Conceptos 4

 1.3 Sistemas que incluyen el proceso Seguridad y Salud en el Trabajo. 5

 1.3.1 Sage ABRA HRMS 5

 1.3.2 SAP ERP HCM 5

 1.3.3 PeopleSoft 6

 1.3.4 EXACT 6

 1.4 Modelo de desarrollo de Software 8

 1.5 Marco de trabajo. 10

 1.6 Herramientas para el desarrollo. 11

 1.6.1 Herramienta para el modelado 11

 1.6.2 Ambiente de desarrollo integrado (IDE)..... 11

 1.6.3 Herramienta para el desarrollo colaborativo 12

 1.6.4 Lenguajes de programación 12

 1.6.5 Gestor de Base de Datos 12

 1.6.6 Servidor Web 13

 1.6.7 Navegador Web..... 13

 1.7 Conclusiones parciales 13

2. SOLUCIÓN PROPUESTA..... 14

 2.1 Introducción 14

 2.2 Descripción del sistema 14

 2.3 Modelado del negocio 14

 2.3.1 Mapa de procesos..... 14

 2.3.2 Procesos de Negocio 15

 2.3.3 Modelo conceptual..... 16

 2.4 Especificación de los Requisitos del Sistema 17

 2.4.1 Requisitos Funcionales 18

 2.4.2 Descripción de los Requisitos del Sistema 22

 2.4.3 Técnicas para la validación de los requisitos 28

 2.4.4 Requisitos No Funcionales 29

 2.5 Conclusiones parciales 31

3. DISEÑO DEL SISTEMA..... 32

 3.1 Introducción 32

 3.2 Patrón Arquitectónico Utilizado 32

 3.3 Patrones de Diseño Utilizados 33

3.3.1 Patrones GOF	34
3.3.2 Patrones GRASP	34
3.4 Diagramas de clases del diseño	36
3.5 Diagramas de secuencia	37
3.6 Modelo de datos	38
3.7 Diagrama de componentes	39
3.8 Conclusiones parciales	40
4. IMPLEMENTACIÓN Y PRUEBA	41
4.1 Introducción	41
4.2 Diagrama de despliegue	41
4.3 Funcionalidades implementadas	42
4.4 Métricas para la validación del diseño propuesto	43
4.4.1 Métrica Tamaño operacional de clase	44
4.4.2 Métrica Relaciones entre clases	49
4.5 Pruebas de software	54
4.5.1 Pruebas de caja blanca	55
4.5.2 Pruebas de caja negra	59
4.6 Conclusiones parciales	66
CONCLUSIONES.....	67
RECOMENDACIONES.....	68
REFERENCIAS BIBLIOGRÁFICAS.....	69
ANEXOS.....	71
Anexo 1: Acta de liberación	71
Anexo 2: Descripción del proceso Riesgos Laborales	73
Anexo 3: Diagrama de procesos de Registro de medios de protección	75
Anexo 4: Diagrama de procesos de Capacitación en SST	77
Anexo 4: Diagrama de procesos de Atención médica	79
Anexo 5: Diagrama de procesos de Registro de los accidentes de trabajo	81
Anexo 6: Diagramas de clases del diseño	83
Anexo 7: Diagramas de secuencia	86
GLOSARIO	91

INTRODUCCIÓN

La gestión del capital humano constituye un factor necesario para el desarrollo y la evolución de las compañías en el mundo, el cual se puede resumir como la organización y control del personal que individual y colectivamente contribuyen a alcanzar los objetivos y las metas de cualquier organización (1).

Actualmente, las entidades necesitan innovar y trabajar con eficiencia para poder reaccionar rápidamente a los cambios del mercado y a la competencia, para lograrlo es necesario entre otros elementos, adoptar sistemas informáticos que realicen tareas de análisis, de gestión de la información y del personal que las componen.

Como parte de los sistemas informáticos empleados por las empresas podemos destacar los de gestión empresarial, que son aplicaciones informáticas utilizadas para garantizar una administración más efectiva, con el fin de volver más eficiente el proceso productivo de las mismas (2).

Dentro de los más destacados podemos nombrar a los Sistemas de Planificación de Recursos Empresariales (ERP, por sus siglas en inglés), los cuales están preparados para adaptarse a los cambios de una organización y como herramientas tienen mejoras tanto técnicas como funcionales (3).

La funcionalidad de los ERP se encuentra dividida por módulos independientes, pero con la facilidad de interactuar entre ellos, según las necesidades del cliente. De esta forma se ahorra tiempo y se minimizan los errores, al no existir dependencia entre las operaciones de la empresa. Una de las necesidades por las que hoy día se exige la adquisición de un sistema con estas características, es porque el mercado se hace más grande, además proporcionan mayor integración de la información y modernización de los procesos de negocio que le permitirá a las entidades mayor eficiencia (3).

En Cuba, es una necesidad elevar la productividad y la eficiencia de las empresas, contribuyendo de esta forma al perfeccionamiento de las mismas. Un buen ambiente laboral influye positivamente en la motivación para realizar las tareas y la destreza con que se ejecute el trabajo. Se puede afirmar que aunque exista mucho cuidado por parte de la empresa en la selección de los candidatos más idóneos, en su capacitación para el puesto y crear una atmósfera óptima de trabajo, si las condiciones físicas de la entidad son inadecuadas, la producción mermará y ello ocasionará efectos negativos, como son: disminución de la productividad, aumento de errores, mayor índice de accidentes y mayor fluctuación de personal (4).

Uno de los requisitos establecidos para las empresas dentro del llamado Perfeccionamiento Empresarial² es la Seguridad y Salud en el Trabajo, como un proceso del Sistema de Gestión de Capital Humano, que debe cumplir las regulaciones del Ministerio del Trabajo y Seguridad Social, así como las normas obligatorias aprobadas por la Oficina Nacional de Normalización de la República de Cuba sobre este tema (2).

Seguridad y Salud en el Trabajo (SST), es el proceso orientado a crear las condiciones, capacidades y cultura de prevención para que el trabajador y su organización desarrollen la labor eficientemente y sin riesgos, evitando sucesos que originen daños derivados del trabajo, que puedan afectar la salud e integridad del trabajador, el patrimonio de la organización y el medio ambiente (1).

El proceso de SST en las entidades presupuestadas y empresariales del gobierno cubano se lleva a cabo de forma manual, archivando la información en papel, lo cual provoca duplicación de la misma, pérdida de datos y que los reportes estadísticos no sean los más exactos; de esta forma el desarrollo de este proceso se torna complejo y engorroso, lo que conlleva al mal funcionamiento de las entidades en general.

Actualmente en Cuba se encuentra en etapa de desarrollo, en la Universidad de las Ciencias Informáticas el primer ERP cubano, cuyo nombre es Cedrux (5). El mismo cuenta con el subsistema Capital Humano que abarca un grupo de procesos que interactúan entre sí y permiten que se gestione de forma correcta la organización del trabajo, la selección e integración del personal, la evaluación del desempeño, la estimulación moral y material, entre otros, pero en estos momentos no se cuenta con un componente que garantice el adecuado manejo de la seguridad y la salud en el trabajo.

Lo antes expuesto permite plantear el siguiente **problema a resolver**: En el subsistema Capital Humano del Sistema Integral de Gestión Cedrux, no se permite la gestión de la seguridad y salud en el trabajo como establecen las normas vigentes en el país para las entidades empresariales presupuestadas.

Se define como objeto de estudio la gestión de la Seguridad y Salud en el Trabajo, enmarcando como **campo de acción** los sistemas que gestionan el proceso de Seguridad y Salud en el Trabajo en las entidades empresariales y presupuestadas.

Para darle solución al problema planteado se define como **objetivo general**: Desarrollar un componente de Seguridad y Salud en el Trabajo para el subsistema Capital humano del Sistema Integral de Gestión Cedrux que permita la gestión de este proceso como lo establecen las normas vigentes en el país para las entidades empresariales y presupuestadas.

²Sistema de dirección y gestión empresarial que tiene como objetivo central incrementar al máximo la eficiencia y competitividad de las empresas estatales (21).

Desglosado en los siguientes **objetivos específicos**:

- Realizar el diseño teórico metodológico de la investigación que permita conocer los elementos que la componen y las características referentes al proceso Seguridad y Salud en el Trabajo.
- Realizar la captura de los requisitos para el proceso Seguridad y Salud en el trabajo como parte del subsistema Capital humano.
- Realizar el análisis y diseño para el proceso Seguridad y Salud en el trabajo.
- Implementar las funcionalidades diseñadas.
- Validar los resultados obtenidos en las fases antes mencionadas.

Concretándose como **idea a defender**: El desarrollo de un componente de seguridad y salud en el trabajo del subsistema Capital humano de Cedrux permite la gestión de este proceso como lo establecen las normas vigentes en el país para las entidades empresariales y presupuestadas.

La presente investigación está estructurada en cuatro capítulos y varios anexos, en los cuales se tratan las siguientes cuestiones:

Capítulo 1: Fundamentación Teórica. Se presenta el marco teórico estudiado y seleccionado para desarrollar el presente trabajo, orientado al desarrollo de software para la gestión de la seguridad y salud en el trabajo. Se muestra una panorámica sobre los conceptos y herramientas necesarios para alcanzar los objetivos planteados.

Capítulo 2: Solución Propuesta. Se realiza la descripción de la solución propuesta.

Capítulo 3: Diseño del sistema. Se lleva a cabo el diseño del componente. Se describirán los patrones de diseño empleados.

Capítulo 4: Implementación y prueba. Se describe todo lo relacionado con la implementación del componente, así como con las pruebas realizadas al mismo.

1. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se abordan los principales conceptos relacionados con el proceso Seguridad y Salud en el Trabajo que ayudan a comprender la solución propuesta. Se caracterizan algunos de los sistemas existentes que incluyen el proceso y se realiza la descripción de las herramientas y metodologías a utilizar para desarrollar este trabajo de diploma.

1.2 Principales Conceptos

La Administración de Recursos Humanos (ARH) es una especialidad que surgió con el crecimiento de las organizaciones y con la complejidad de las tareas organizacionales. Consiste en la planeación, en la organización, en el desarrollo y en la coordinación y control de técnicas capaces de promover el desempeño eficiente del personal, a la vez que la organización represente el medio que permite a las personas que colaboran en ella, alcanzar los objetivos individuales relacionados directos o indirectamente con el trabajo(1).

Desde el punto de vista de la ARH, la salud y la seguridad de los empleados constituyen una de las principales bases para la preservación de la fuerza de trabajo adecuada. De modo genérico, higiene y seguridad en el trabajo constituyen dos actividades íntimamente relacionadas para garantizar condiciones personales y materiales de trabajo capaces de mantener cierto nivel de salud de los empleados (1).

Seguridad en el Trabajo es el conjunto de medidas técnicas, educacionales, médicas y psicológicas, empleadas para prevenir los accidentes, eliminar las condiciones inseguras del ambiente, e instruir o convencer a las personas sobre la implantación de medidas preventivas. Su empleo es indispensable para el desarrollo satisfactorio del trabajo (1).

El proceso Seguridad y Salud en el Trabajo tiene como objetivo fundamental preservar la vida y salud de los trabajadores y para ello debe preverse en los programas de prevención de riesgos laborales que se establezcan los instrumentos, procedimientos e investigaciones encaminadas a disminuir la accidentalidad en la esfera laboral (6).

SST constituye una parte fundamental dentro de la gestión del capital humano en una empresa, pues el mismo tiene como propósito crear el ambiente adecuado para que los trabajadores desarrollen su labor sin riesgos, evitando de esta forma eventos que puedan ocasionar daños a su salud e integridad. Por tanto, es de suma

importancia para el patrimonio de la empresa, la familia del trabajador y la sociedad en general que se gestione la SST para evitar, prevenir y mitigar la ocurrencia de accidentes que pueden o no producir daños al personal de la entidad.

La investigación de los accidentes de trabajo y sus causas es un factor importante para evitar la repetición de eventos similares. Es necesario establecer la obligatoriedad del registro, investigación e información de los accidentes de trabajo, en todas las entidades laborales, acorde a principios y procedimientos actualizados y adecuados, con el objetivo de obtener datos uniformes y suficientes que permitan orientar las estrategias de prevención específicas y evaluar su comportamiento en empresas, ramas y sectores, así como para realizar comparaciones internacionales (6).

1.3 Sistemas que incluyen el proceso Seguridad y Salud en el Trabajo.

Existen algunos sistemas que integran y manejan los procesos de negocio de las empresas, dentro de los cuales se encuentra el proceso de Seguridad y Salud en el trabajo, por lo que es importante valorar sus particularidades.

1.3.1 Sage ABRA HRMS

Es una herramienta para la gestión de la información del empleado. Sirve a empresas de cualquier tamaño y ayuda a administrar los beneficios de las mismas con herramientas completas de administración (5).

Dentro de sus módulos, Seguridad y Salud en el Trabajo permite:

- Crear fácil y rápidamente los ahorros exactos y los planes de seguro.
- Identificar las áreas problemáticas para ayudar a mejorar la seguridad y cumplir con las Regulaciones de Salud relativas a los accidentes y la indemnización de los trabajadores y da seguimiento a las reclamaciones.

Sin embargo no es una propuesta viable para la economía del país, debido a que se incurre en costosos gastos por los conceptos de licencia y mantenimiento de software, ya que las tecnologías que se utilizan para el desarrollo y sustento del sistema son propietarios.

1.3.2 SAP ERP HCM

Los últimos informes de los analistas confirman que SAP lidera el mercado de software de Gestión del Capital Humano. Ayuda a maximizar el potencial de los empleados en múltiples industrias de todo el mundo, ofrece una

solución de gestión con una capacidad global y permite a las organizaciones las herramientas necesarias para gestionar su activo más importante: las personas. La solución ayuda a los líderes a pronosticar, planificar y contratar a los mejores talentos, así como cultivar las habilidades y formar a sus trabajadores(5).

Uno de sus módulos es Seguridad y Salud en el Trabajo, el mismo:

- Gestiona la higiene industrial, salud ocupacional, el cumplimiento de la seguridad de productos, sustancias peligrosas, mercancías peligrosas y la gestión de residuos.

SAP ERP no es una solución factible al problema que se expone, debido a que el país no cuenta con los recursos financieros suficientes para adquirirlo y sus licencias tienen un alto valor monetario. Además de que es un software extranjero y desarrollado para plataforma de software propietario.

1.3.3 PeopleSoft

Oracle'sPeopleSoft Enterprise Human Resources, provee un conjunto de módulos estratégicos que ayudan a obtener ventajas competitivas a la organización. Este producto permite encontrar la persona correcta para la compañía, y llevar el progreso profesional de los empleados, garantizando al mismo tiempo que los salarios y las prácticas de empleo sigan siendo justos y competitivos (5).

Este software incluye como uno de sus módulos Seguridad y Salud en el Trabajo que permite:

- Llevar un registro de los datos de certificados médicos, seguros, informes de incidencias y condiciones laborales por cada puesto de trabajo.

Es un software privativo adaptable solo para plataformas de Windows. Incluye soporte para la gestión empresarial en algunos países en los que no está incluida Cuba.

1.3.4 EXACT

Es una solución web que permite integrar, administrar, controlar y mejorar la forma de trabajo de todo el personal, así como añadir valor a través del negocio (5).

Dentro de sus módulos, Seguridad y Salud en el Trabajo es el encargado de:

- Llevar un seguimiento de los datos de las condiciones de trabajo y la salud de los empleados, incluidos los informes de incidentes detallados, la enfermedad y datos de lesiones por los individuos.
- Controlar los planes de rehabilitación y terapia, el diagnóstico médico, información de seguros y reclamaciones, resultados de exámenes médicos relacionados con las lesiones de trabajo o enfermedad.

- Llevar un seguimiento de la información personal, tipo de sangre, nombre del médico, dirección, restricciones laborales, audiometría y la salud respiratoria.

Aunque es un sistema avanzado con respecto a las exigencias funcionales que requiere el país, los gastos en cuestiones de licencia y mantenimiento del software son muy elevados, por tanto no es una solución viable para la economía del país.

En lo que a Salud y Seguridad en el Trabajo se refiere, en la mayoría de los casos no se trata de funcionalidades orientadas a la evaluación y actualización de los riesgos; así como los planes de medidas para su solución. Se observa que algunos de los sistemas poseen funcionalidades requeridas por el Sistema de Gestión Integrada de Capital Humano en Cuba, pero no recogen todos los requisitos de las Normas Cubanas 3001:2007 y 3002:2007. En la siguiente tabla se resumen las características deben cumplir los sistemas analizados para ser utilizados.

Características deseadas	Sistemas			
	Sage ABRA HRMS	SAP ERP HCM	PeopleSoft	EXACT
Cumple con las normas cubanas.	NO	NO	NO	NO
Cumple con el paradigma de independencia tecnológica del país.	NO	NO	NO	NO
Realiza la gestión de las situaciones peligrosas y los riesgos asociados.	SI	NO	NO	NO
Trata la atención médica de los trabajadores.	NO	SI	SI	SI
Registra y controla los medios de protección individuales y colectivos	NO	NO	NO	NO

Tabla 1: Características que deben cumplir los sistemas.

Luego de estudiar y analizar los sistemas anteriores se puede concluir que ninguno cumple con todas las características deseadas, por lo que se cree necesario el desarrollo de un sistema que responda a las particularidades de la economía cubana, que incluya las características positivas de los sistemas estudiados y sea consecuente con el paradigma de independencia tecnológica por el cual apuesta el país.

1.4 Modelo de desarrollo de Software

El Centro de Informatización de la Gestión de Entidades (CEIGE), de la Universidad de las Ciencias Informáticas (UCI), centro al cual pertenece el equipo de trabajo, propone un modelo de desarrollo de software que detalla el ciclo de vida de sus proyectos.

El modelo utilizado incorpora los subprocesos dictados por el Nivel II de CMMI (Capability Maturity Model Integration) (7).

A continuación se brindan algunas características de este modelo:

- **Centrado en la arquitectura:** La arquitectura es una vista del diseño completa con las características más importantes, dejando a un lado los detalles. Ésta no sólo incluye las necesidades de los usuarios e inversores, sino también otros aspectos técnicos como el hardware, sistema operativo, sistema de gestión de base de datos, protocolos de red; con los que debe coexistir el sistema. La arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas (8).
- **Orientado a componentes:** Nos lleva a alcanzar un mayor nivel de reutilización de software, aún en contextos distintos a aquellos para los que fue diseñado. Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados. Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema. Dado que un componente puede ser construido y luego mejorado continuamente, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo (8).
- **Iterativo e incremental:** La alta complejidad de los sistemas actuales hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos. Cada uno de estos mini-proyectos se les denomina iteración y pueden o no representar un incremento en el grado de terminación del producto completo. En cada iteración los desarrolladores seleccionan un grupo de casos de uso, los cuales se diseñan,

implementan y prueban. La planificación de iteraciones hace que se reduzcan los riesgos de los costes de un solo incremento, no sacar al mercado un producto en el tiempo previsto, mantener la motivación del equipo pues puede ver avances claros a corto plazo y que el desarrollo pueda adaptarse a los cambios en los requisitos (8).

- **Ágil y adaptable al cambio:** Los clientes y funcionales son involucrados en el proyecto, por lo que poseen parte de la responsabilidad del éxito del mismo. Semanalmente se concilian, discuten y aprueban los cambios. El desarrollo de las partes formaliza solo las características principales de la solución, priorizándose de esta manera los talleres y las comunicaciones (8).

A continuación se detallan las disciplinas y los artefactos a generar siguiendo el modelo, teniendo en cuenta la estructura del presente trabajo.

Modelado del negocio: Es la disciplina destinada a comprender los procesos de negocio de la organización. Se comprende cómo funciona el negocio que se desea automatizar para tener garantías de que el software desarrollado va a cumplir su propósito (8).

- Modelo Conceptual
- Mapa de procesos de Negocio.
- Descripción de procesos de negocio

Requisitos: El esfuerzo principal en la disciplina de Requisitos es desarrollar un modelo del sistema que se va a construir. Incluye un conjunto de artefactos que describen todas las interacciones que tendrán los usuarios con el software y que responden a los requisitos funcionales del sistema. Se especifican los requisitos funcionales y no funcionales (8).

- Especificación de requisitos de software
- Descripción de requisitos
- Documento de salidas del sistema (actualizado)

Análisis y diseño: Durante esta disciplina es modelado el sistema para que soporte todos los requisitos. Esto contribuye a una arquitectura sólida y estable que se convierte en un plano para la próxima. Los artefactos generados en esta etapa son más formales y específicos de una implementación. En caso de llevarse a cabo la reutilización de componentes software ya desarrollados, durante esta disciplina se ajusta el modelado existente a los requisitos actuales (8).

- Diagrama de clases del diseño por componentes o por módulos

- Diagrama de interacción
- Modelo de datos
- Diagrama de componentes

Implementación: A partir de los resultados del análisis y diseño se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares. Al reutilizar componentes software ya implementados se lleva a cabo el desarrollo necesario para ajustar a los requisitos actuales y posteriormente realizar la integración de los componentes (8).

- Ficheros de código
- Prototipos de interfaz funcionales.
- Diagrama de despliegue.

Pruebas internas: Durante esta disciplina se desarrollan las pruebas del grupo de calidad del centro verificando el resultado de la implementación. Permite identificar posibles errores en la documentación y el software, es decir requisitos que el producto debería cumplir y que aún no los cumple (8).

- Diseño de casos de prueba
- Documento de no conformidades
- Acta de liberación

1.5 Marco de trabajo.

El Departamento de Tecnología del centro de desarrollo CEIGE cuenta con un marco de trabajo denominado Sauxe, el cual se utiliza para el desarrollo de los proyectos del centro. Por ser este marco el definido por la arquitectura del proyecto se utilizará para el desarrollo de esta investigación.

Sauxe cuenta con una arquitectura en capas que a su vez presenta en su capa superior un Modelo Vista Controlador³. Contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo (9).

Las tecnologías libres con las que cuenta el marco de trabajo son:

³ Modelo Vista Controlador es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Zend_Framework 1.7: Se trata de un framework para desarrollo de aplicaciones Web y servicios Web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Además es de código abierto y trabaja con PHP 5. Está formado por una serie de métodos estáticos y varios componentes que se pueden reutilizar (9).

Doctrine 1.2.1: Potente y completo sistema de mapeo relacional de objetos (por sus siglas en inglés ORM) para PHP 5.2 con una capa de abstracción de bases de datos incorporado. Brinda la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos (10).

Extjs 2.2: Es una biblioteca o conjunto de librerías de JavaScript para el desarrollo de aplicaciones web interactivas, usa tecnologías AJAX2, DHTML3 y DOM4. Permite realizar completas interfaces de usuario, fáciles de usar, muy parecidas a las conocidas aplicaciones de escritorio, posibilitando a los desarrolladores concentrarse en las funcionalidades de las aplicaciones (11).

1.6 Herramientas para el desarrollo.

A continuación se describen varias herramientas que por sus características y ventajas fueron definidas por el CEIGE y por consiguiente serán las utilizadas en el desarrollo de la solución técnica.

1.6.1 Herramienta para el modelado

Visual Paradigm 6.4: Se utilizará el Visual Paradigm para visualizar y diseñar los elementos de software debido a que es multiplataforma y por las facilidades que brinda para el diseño de los diagramas necesarios y su documentación. Soporta los principales estándares como UML, SysML, BPMN y XML. Ofrece un completo conjunto de herramientas a los equipos de desarrollo de software, necesarios para la captura de requisitos, la planificación de software, la planificación de pruebas, el modelado de clases y el modelado de datos (12).

1.6.2 Ambiente de desarrollo integrado (IDE)

Netbeans IDE 7.0.1: Es un reconocido entorno de desarrollo integrado, disponible para Windows, Mac, Linux y Solaris. El proyecto Netbeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles

utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy and Grails y C/C++ (12).

1.6.3 Herramienta para el desarrollo colaborativo

SVN (Subversión) 1.6.6: Es un software de sistema de control de versiones, libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que los archivos versionados tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo (13).

1.6.4 Lenguajes de programación

JavaScript: Es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas (14).

PHP 5.2.6: Lenguaje de programación generalmente usado en la programación de sitios web dinámicos y actualmente es casi el lenguaje de desarrollo de sitios más usado en todo el mundo. Es un potente lenguaje interpretado, solo se necesita un navegador web para ejecutarlo, es un lenguaje del lado del servidor, por lo que los scripts se ejecutan remotamente y el resultado aparece en la máquina cliente(local), tiene soporte para muchos tipos de bases de datos, entre las principales están MySQL, PostgreSQL, SQLite, entre otras (15).

1.6.5 Gestor de Base de Datos

PostgreSQL 8.3: Es un servidor de base de datos objeto relacional libre, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, etc. La documentación está muy bien organizada, pública y libre. Posee bajo costo de propiedad total y rápido retorno de la inversión (16).

1.6.6 Servidor Web

Apache 2.2.9: Es un servidor gratuito de código fuente abierto, potente y que ofrece un servicio estable y sencillo de mantener y configurar. Se ejecuta en varios Sistemas Operativos, haciéndolo esta característica prácticamente universal. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor (17).

1.6.7 Navegador Web

Mozilla Firefox 2.17: Presenta una forma rápida y eficiente de navegar por la web, que permite abrir varias páginas en una misma ventana mediante el empleo de pestañas separadas. Es un navegador multiplataforma y está disponible en varias versiones de Microsoft Windows, Mac OS X, GNU/Linux y algunos sistemas basados en Unix. Su código fuente es software libre, publicado bajo una triple licencia GPL/LGPL/MPL (18).

1.7 Conclusiones parciales

- Se hizo una revisión de los sistemas más relevantes por su amplio uso y sus prestaciones, que incluyen el proceso Seguridad y Salud en el Trabajo, demostrando la necesidad que existe del desarrollo de un componente de SST en el subsistema Capital Humano del Sistema Cedrux, que responda a las características particulares de las entidades cubanas y que incluya los aspectos positivos de los sistemas estudiados.
- Se describen las tecnologías, el modelo de desarrollo y herramientas que por sus características son adecuadas para la solución, cumpliendo de esta manera con el paradigma de independencia tecnológica por el cual apuesta el país y ajustándose a lo definido por el departamento de tecnología del CEIGE para Cedrux.

2. SOLUCIÓN PROPUESTA

2.1 Introducción

En el presente capítulo se brinda una descripción de la solución propuesta, de las entradas y salidas del proceso Seguridad y Salud en el Trabajo a través del mapa de procesos, se realiza el modelo conceptual. Se muestran los requisitos funcionales y no funcionales.

2.2 Descripción del sistema

El sistema que se propone es una aplicación web que garantiza el manejo de la Seguridad y Salud en el Trabajo acorde a las Normas Cubanas 3001:2007, 3002:2007 y la resolución No. 31/ 2002.

Cuenta con funcionalidades que se encargan de gestionar, evaluar y controlar los riesgos laborales presentes en las áreas y puestos de trabajo. Brinda la posibilidad de registrar y asignar los medios de protección por trabajador y puesto de trabajo, permite planificar y asignar las actividades de capacitación de SST. Mediante el sistema se obtiene un control de la historia clínica y los datos de salud del trabajador para que de esta forma se puedan planificar chequeos médicos, además permite investigar, registrar y controlar los accidentes de trabajo.

2.3 Modelado del negocio

2.3.1 Mapa de procesos

El mapa de procesos ofrece una visión general del sistema, es una representación gráfica de todos los procesos que lo componen y de la relación que existe entre ellos (19).

Para el componente de Seguridad y Salud en el Trabajo se definieron 5 procesos fundamentales: Riesgos laborales, Registro de medios de protección, Capacitación en SST, Atención médica y Registro de los accidentes de trabajo, los cuales interactúan con otros componentes a la hora de ofrecer y recibir la información necesaria para garantizar el funcionamiento de distintas funcionalidades.

A continuación se muestra el mapa de procesos del componente para un mayor entendimiento del negocio.

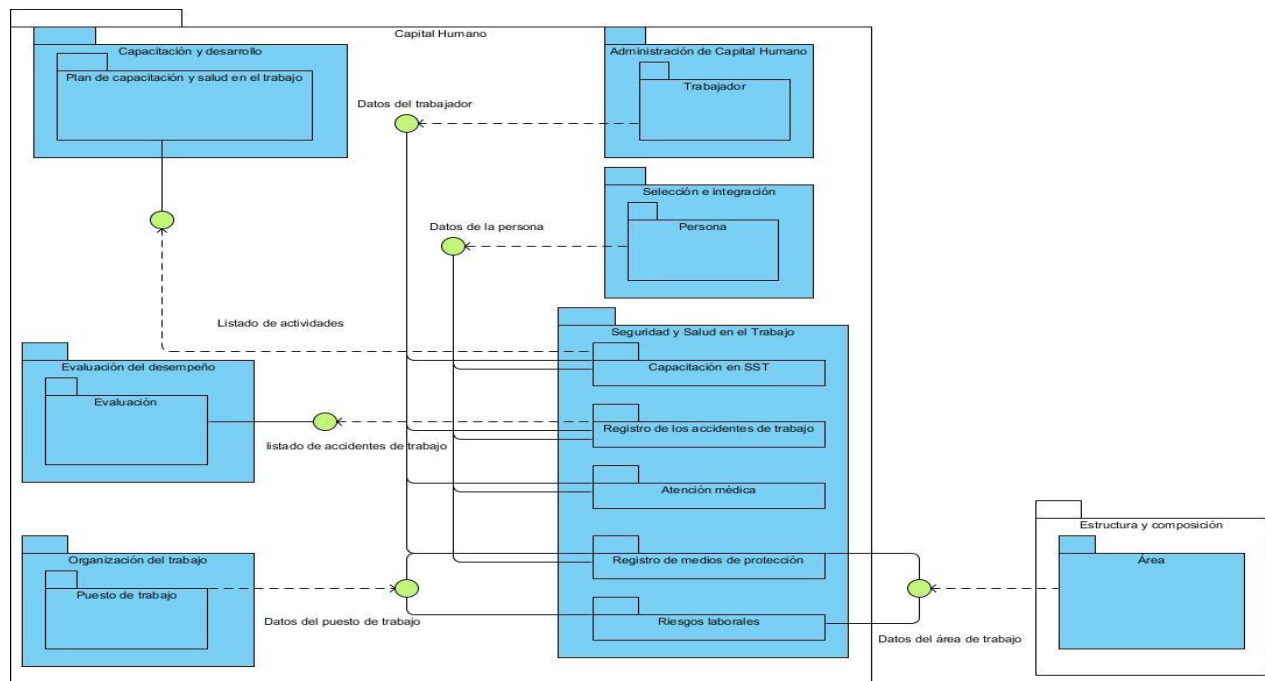


Figura 1: Mapa de procesos de Seguridad y Salud en el Trabajo.

2.3.2 Procesos de Negocio

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de un negocio definido. Cada proceso de negocio tiene sus entradas, funciones y salidas. Puede ser parte de un proceso mayor que lo abarque o bien puede incluir otros procesos de negocio (20).

El proceso Riesgos laborales tiene como objetivo identificar las situaciones peligrosas, evaluar y controlar los riesgos asociados a ellas, así como la elaboración del plan de medidas para cada riesgo existente en la organización. El resto de los procesos y sus descripciones detalladas se pueden encontrar en los anexos.

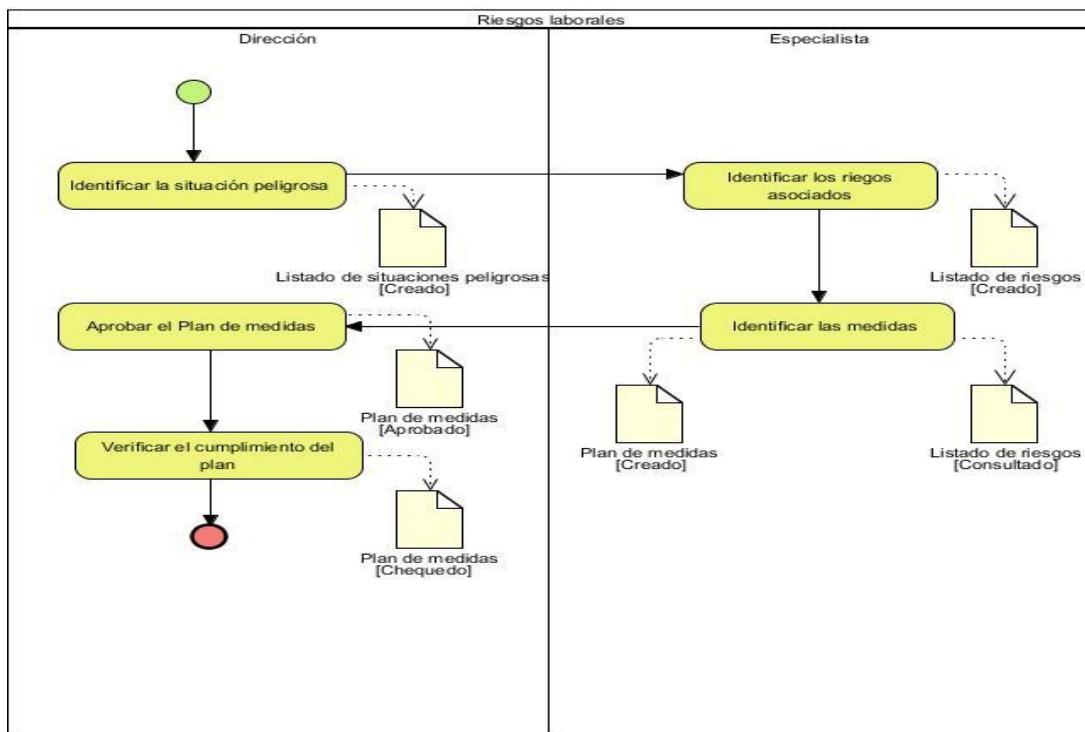


Figura 2: Diagrama de procesos de Riesgos laborales.

2.3.3 Modelo conceptual

Los modelos conceptuales se utilizan para representar la realidad a un alto nivel de abstracción. El objetivo de la creación del modelo conceptual es aumentar la comprensión del problema y contribuir a esclarecer la terminología o nomenclatura del dominio (21).

En la Figura 3: **Modelo conceptual** se muestra el modelo conceptual que se generó para el proceso Seguridad y Salud en el Trabajo.

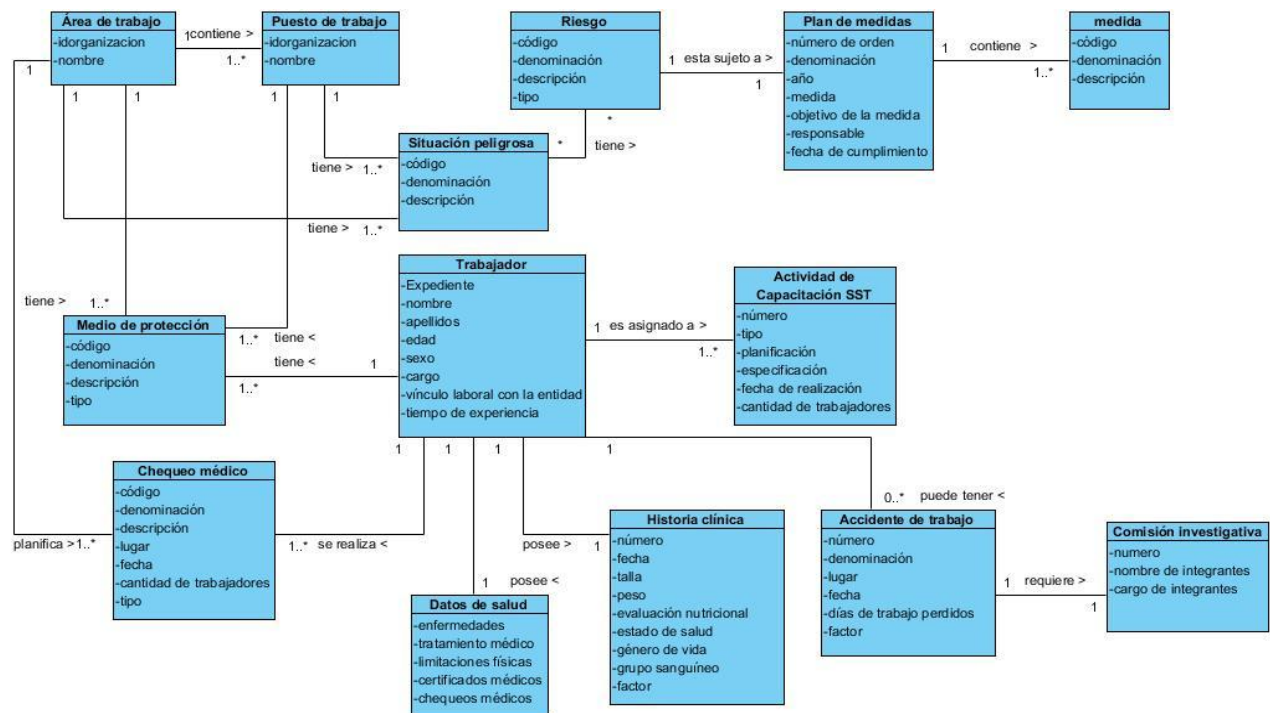


Figura 3: Modelo conceptual del proceso Seguridad y Salud en el Trabajo.

2.4 Especificación de los Requisitos del Sistema

Un requisito es la condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado. Los requisitos de un sistema describen los servicios que han de ofrecer y las restricciones asociadas a su funcionamiento.

La captura de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema (22).

A continuación se presentan un grupo de técnicas que se utilizaron para identificar los requisitos del sistema:

Entrevista: Técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y documentación de los resultados (22).

Tormentas de ideas: Técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. Como técnica de captura de requisitos es sencilla de usar y de aplicar, puesto que no requiere tanto trabajo en grupo como éste. Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del mismo, por lo que suele aplicarse en los primeros encuentros (22).

Revisión de la documentación: La utilización de esta técnica depende de la información que se tenga almacenada acerca del software que se desee construir. Las entidades guardan información acerca de sus procesos, los modelos o informes necesarios para su desarrollo o para rendir cuenta a los organismos superiores (22).

2.4.1 Requisitos Funcionales

Los requisitos funcionales definen qué debe hacer un sistema (22). Para la elaboración del proceso se obtuvo 20 agrupaciones de requisitos para un total de 110 requisitos funcionales. A continuación se muestra una selección de los mismos:

1. Gestionar situaciones peligrosas	Adicionar situación peligrosa
	Modificar situación peligrosa
	Eliminar situación peligrosa
	Listar situaciones peligrosas
	Buscar situación peligrosa
2. Gestionar riesgos laborales	Adicionar riesgos laborales
	Modificar riesgos laborales
	Eliminar riesgos laborales
	Listar riesgos laborales
	Buscar situación peligrosa
	Buscar riesgos laborales
	Listar situación peligrosa

3. Gestionar plan de medidas	Adicionar plan de medidas
	Modificar plan de medidas
	Eliminar plan de medidas
	Listar plan de medidas
	Buscar plan de medidas
	Listar riesgos laborales
	Buscar riesgos laborales
	Chequear plan de medidas
4. Registrar medios de protección por puesto de trabajo	Adicionar medio de protección
	Modificar medio de protección
	Eliminar medio de protección
	Listar medios de protección
	Buscar medios de protección
	Listar áreas y puestos de trabajo
5. Asignar medios por puesto de trabajo	Buscar medios de protección
	Listar medios de protección
	Adicionar medio de protección
	Modificar medio de protección
	Eliminar medio de protección
	Listar áreas y puestos de trabajo
6. Asignar medios al trabajador	Buscar medios de protección
	Listar medios de protección
	Buscar trabajador
	Adicionar medio al trabajador
	Modificar medio al trabajador
	Eliminar medio al trabajador
7. Gestionar actividades de capacitación de SST	Adicionar actividad
	Modificar actividad

	Eliminar actividad
	Listar actividad
	Buscar actividad
8. Registrar trabajadores por actividad de capacitación	Buscar trabajador
	Buscar actividad
	Vincular trabajadores
	Desvincular trabajadores
	Listar actividad
	Listar trabajador
9. Registrar historia clínica del trabajador	Adicionar historia clínica
	Modificar historia clínica
	Eliminar historia clínica
	Listar historia clínica
10. Registrar datos de salud del trabajador	Adicionar datos de salud
	Modificar datos de salud
	Eliminar datos de salud
	Listar datos de salud
11. Planificar chequeos médicos al trabajador	Adicionar chequeo
	Modificar chequeo
	Eliminar chequeo
	Listar chequeos
	Buscar chequeo
	Verificar Plan de chequeos médicos
12. Planificar chequeos médicos por área de trabajo	Adicionar chequeo
	Modificar chequeo
	Eliminar chequeo
	Listar chequeos

	Buscar chequeo
	Listar áreas y puestos de trabajo
13. Registrar accidente de trabajo al trabajador	Adicionar accidente al trabajador
	Modificar accidente al trabajador
	Eliminar accidente al trabajador
	Listar accidente al trabajador
	Buscar accidente al trabajador
	Evaluar accidente al trabajador
14. Registrar accidente de trabajo	Adicionar accidente al trabajador
	Modificar accidente al trabajador
	Eliminar accidente al trabajador
	Listar accidente al trabajador
	Buscar accidente al trabajador
15. Gestionar Nomenclador Medida a ejecutar	Adicionar medida
	Modificar medida
	Eliminar medida
	Listar medida
	Buscar medida
16. Gestionar Nomenclador Tipos de riesgo	Adicionar Tipos d riesgo
	Modificar Tipos de riesgo
	Eliminar Tipos de riesgo
	Listar Tipos de riesgo
	Buscar Tipos de riesgo
17. Gestionar Nomenclador objetivo de la medida	Adicionar objetivo de la medida
	Modificar objetivo de la medida
	Eliminar objetivo de la medida
	Listar objetivo de la medida

	Buscar objetivo de la medida
18. RF: Gestionar Nomenclador tipo de chequeo médico	Adicionar tipo de chequeo médico
	Modificar tipo de chequeo médico
	Eliminar tipo de chequeo médico
	Listar tipo de chequeo médico
	Buscar tipo de chequeo médico
19. RF: Gestionar Nomenclador medios de protección	Adicionar medios de protección
	Modificar medios de protección
	Eliminar medios de protección
	Listar medios de protección
	Buscar medios de protección
20. RF: Gestionar Nomenclador factores que originaron el accidente	Adicionar factores
	Modificar factores
	Eliminar factores
	Listar factores
	Buscar factores

2.4.2 Descripción de los Requisitos del Sistema

A continuación se muestra de ejemplo la descripción del requisito Gestionar situaciones peligrosas.

Descripción textual del requisito Adicionar situación peligrosa.

Precondiciones	Se ha registrado al menos un puesto de trabajo en el sistema.
Flujo de eventos	
Flujo básico	
1	Se señala el puesto de trabajo al cual se le registrará la situación peligrosa.
2	Se introducen los datos de la situación peligrosa: Código: Es el identificador de la situación peligrosa. Denominación: Es el nombre que recibe la situación peligrosa que se desea crear.

	Descripción: Especifica en qué consiste la situación peligrosa.	
3	El sistema valida los datos introducidos.	
4	Si los datos son correctos el sistema los registra.	
5	El sistema confirma el registro de los datos.	
6	Concluye el requisito.	
Pos-condiciones		
1	Se registró en el sistema una nueva situación peligrosa.	
Flujos alternativos		
Flujo alternativo 3.a Información errónea		
1	El sistema señala los datos erróneos y permite corregirlos.	
2	El usuario corrige los datos.	
3	Volver al paso 2 del flujo básico.	
Pos-condiciones		
1	NA	
Flujo alternativo 3.b Información incompleta		
1	El sistema señala los datos vacíos y permite corregirlos.	
2	El usuario corrige los datos.	
3	Volver al paso 2 del flujo básico.	
Pos-condiciones		
1	NA	
Flujo alternativo *.a El usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		
1	No se registran los datos.	
Validaciones		
1	Se validan los datos según lo establecido en el Modelo conceptual Seguridad y Salud en el trabajo.	
Relaciones	Requisitos	NA
	Incluidos	

	Extensiones	NA
Conceptos	Puesto de Trabajo	Visibles en la interfaz: Denominación Utilizados internamente: NA.
	Situación peligrosa	Visibles en la interfaz: Código Denominación Descripción Utilizados internamente: NA.
Requisitos especiales	NA	
Asuntos pendientes	NA	

Descripción textual del requisito Modificar situación peligrosa.

Precondiciones	Se ha registrado al menos una situación peligrosa en el sistema.
Flujo de eventos	
Flujo básico	
1	Se selecciona la situación peligrosa a modificar.
2	El sistema muestra y permite editar los datos de la situación peligrosa.
3	Se introducen los datos de la situación peligrosa: Código: Es el identificador de la situación peligrosa. Denominación: Es el nombre que recibe la situación peligrosa que se desea crear. Descripción: Especifica en qué consiste la situación peligrosa.
4	El sistema valida los datos introducidos.
5	Si los datos son correctos el sistema los registra.

6	El sistema confirma el registro de los datos.	
7	Concluye el requisito.	
Pos-condiciones		
1	Se modificaron los datos de la situación peligrosa.	
Flujos alternativos		
Flujo alternativo 4.a Información errónea		
1	El sistema señala los datos erróneos y permite corregirlos.	
2	El usuario corrige los datos.	
3	Volver al paso 4 del flujo básico.	
Pos-condiciones		
1	NA	
Flujo alternativo 4.b Información incompleta		
1	El sistema señala los datos vacíos y permite corregirlos.	
2	El usuario corrige los datos.	
3	Volver al paso 4 del flujo básico.	
Pos-condiciones		
1	NA	
Flujo alternativo *.a El usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		
1	No se registran las modificaciones realizadas.	
Validaciones		
1	Se validan los datos según lo establecido en el Modelo conceptual Seguridad y Salud en el trabajo.	
Relaciones	Requisitos	
	Incluidos	NA
	Extensiones	NA

Conceptos	Situación peligrosa	Visibles en la interfaz: Código Denominación Descripción Utilizados internamente: N/A.
Requisitos especiales	NA	
Asuntos pendientes	NA	
Descripción textual del requisito Eliminar situación peligrosa.		
Precondiciones		Se ha registrado al menos una situación peligrosa en el sistema.
Flujo de eventos		
Flujo básico		
1	Se selecciona la situación peligrosa a eliminar.	
2	El sistema verifica que se pueda eliminar la situación peligrosa.	
3	Se solicita confirmación para eliminar la situación peligrosa.	
4	Si el usuario confirma se elimina la situación peligrosa.	
5	El sistema confirma la eliminación.	
6	Concluye el requisito.	
Pos-condiciones		
1	Se eliminó la situación peligrosa.	
Flujos alternativos		
Flujo alternativo *.a El usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		
1	No se elimina la situación peligrosa.	
Validaciones		

1	N/A	
Relaciones	Requisitos Incluidos	N/A
	Extensiones	N/A
Conceptos	Situación peligrosa	Visibles en la interfaz: Código Denominación Descripción Utilizados internamente: N/A.
Requisitos especiales	N/A	
Asuntos pendientes	N/A	
Descripción textual del requisito Listar situación peligrosa.		
Precondiciones	Se ha registrado al menos una situación peligrosa en el sistema.	
Flujo de eventos		
Flujo básico		
1	El sistema muestra un listado de las situaciones peligrosas. Se muestran: Código: Es el identificador de la situación peligrosa. Denominación: Es el nombre que recibe la situación peligrosa que se desea crear.	
2	Concluye el requisito.	
Pos-condiciones		
1	NA	
Flujos alternativos		
Flujo alternativo		
1	NA	
Pos-condiciones		

1	NA	
Validaciones		
1	NA	
Relaciones	Requisitos Incluidos	NA
	Extensiones	NA
Conceptos	Situación peligrosa	Visibles en la interfaz: Código Denominación Utilizados internamente: NA
Requisitos especiales	NA	
Asuntos pendientes	NA	

2.4.3 Técnicas para la validación de los requisitos

La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea. Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de requisitos son correctos (22).

Revisiones: Está técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante (22).

Prototipos de interfaz de usuarios: Mediante esta técnica se logra una versión inicial del sistema demostrando cómo se van a disponer posteriormente los conceptos que intervienen en el mismo. Los prototipos son la manera

en que el cliente pueda apreciar una primera muestra del producto, y dar ideas o recomendaciones que guíen las necesidades que la aplicación deba cumplir (22).

2.4.4 Requisitos No Funcionales

Los requisitos no funcionales, son propiedades que el sistema debe tener, como la fiabilidad, el tiempo de respuesta, la capacidad de almacenamiento entre otros. Los requisitos no funcionales definen cómo debe ser el sistema, son fundamentales para la calidad y el éxito del producto (24).

A continuación se muestran los requisitos no funcionales que define el equipo de trabajo del proyecto Capital Humano y por ende serán usados en la presente investigación para garantizar el funcionamiento óptimo de la aplicación.

Seguridad

- El sistema concederá acceso a cada usuario autenticado solo a las funciones que le estén permitidas, de acuerdo a la configuración del sistema.
- El sistema manejará mecanismos de encriptación para las contraseñas de los usuarios.

Confiabilidad

- El sistema no permitirá la entrada de datos incorrectos.
- El sistema impondrá campos obligatorios para garantizar la integridad de la información que se introduce por el usuario.

Tolerancia ante fallos

- Ante el fallo de una funcionalidad del sistema, el resto de las funcionalidades que no dependen de esta deberán seguir funcionando.
- El sistema permite detectar fallos internos y notificar al usuario de la ocurrencia de estos.

Comprensibilidad

- Todos los mensajes de error del sistema deberán incluir una descripción textual del error.
- Las etiquetas de cada funcionalidad y los campos de cada interfaz tendrán títulos asociados a su función de negocio.

Cognosibilidad

- El idioma de todas las interfaces de la aplicación será el español.
- Los errores cometidos por el usuario les serán notificados y los mensajes incluirán sugerencias de las posibles soluciones.

Operabilidad

- El sistema contará con un menú que permitirá acceder a todas las funcionalidades para entrar los datos y procesar la información.
- El sistema expondrá el menú general en todo momento para que pueda ser utilizado por el usuario en cualquier momento.

Atracción

- El sistema diferenciará los mensajes de información de los mensajes de error y de advertencia valiéndose de distintos íconos para cada tipo.
- La tipografía y colores serán estándares en toda la aplicación.

Rendimiento

- El sistema no excederá los 2 segundos de respuesta al efectuar acciones de cargar un registro (esta cifra no incluye los retardos por concepto de tráfico de red).
- El sistema no excede en un 1 segundo para efectuar acciones de salva de información (esta cifra no incluye los retardos por concepto de tráfico de red).

Flexibilidad

- El sistema permitirá agregar nuevas funcionalidades o modificar alguna existente sin romper la estructura y consistencia de los componentes.

Estabilidad

- Las plataformas de comunicación entre el sistema y sistemas externos deberán abstraer los comportamientos internos de cada sistema.
- Las plataformas de comunicación entre componentes del sistema deberán abstraer los comportamientos internos de cada sistema.

2.5 Conclusiones parciales

- Se define la propuesta de sistema, donde se obtuvo el mapa y los diagramas de procesos del negocio, posibilitando comprender como funcionan los procesos de las entidades que serán necesarios informatizar.
- La definición de los requisitos de la solución, permitió sentar las bases para las restantes fases de los proceso de diseño e implementación.

3. DISEÑO DEL SISTEMA

3.1 Introducción

En el presente capítulo se dará una descripción del estilo arquitectónico utilizado para el desarrollo de la herramienta. Se describen los patrones de diseño empleados, se presentan los diagramas de clases del diseño por componentes. Además se mostrarán los diagramas de secuencia, los diseños de casos de prueba, modelo de datos y el diagrama de componentes.

3.2 Patrón Arquitectónico Utilizado

Un patrón de arquitectura de software es un esquema genérico probado para solucionar un problema particular, el cual es recurrente dentro de un cierto contexto. Este esquema se especifica describiendo los componentes, con sus responsabilidades y relaciones (25).

Para el desarrollo del componente se utilizó el patrón Modelo Vista Controlador (MVC). Patrón de arquitectura de software, el cual separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio (25).

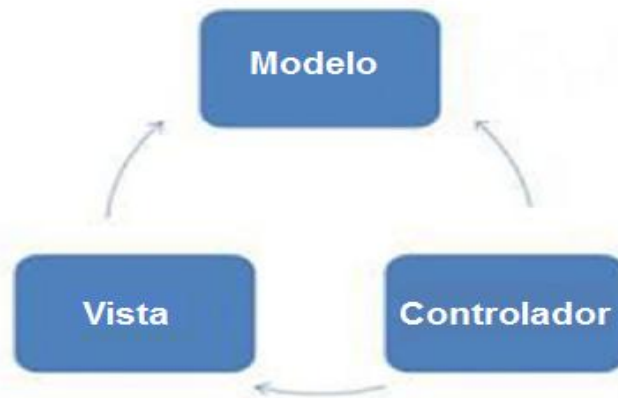


Figura 4: Esquema del patrón Modelo-Vista-Controlador.

El patrón **Modelo – Vista – Controlador** mostrado en la Figura 4: **Esquema del patrón Modelo-Vista-Controlador.**, está catalogado como un patrón de arquitectura de software donde:

Modelo: Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

Vista: Presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.

Controlador: Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

A continuación se muestra un ejemplo de la aplicación de este patrón arquitectónico en el componente desarrollado.

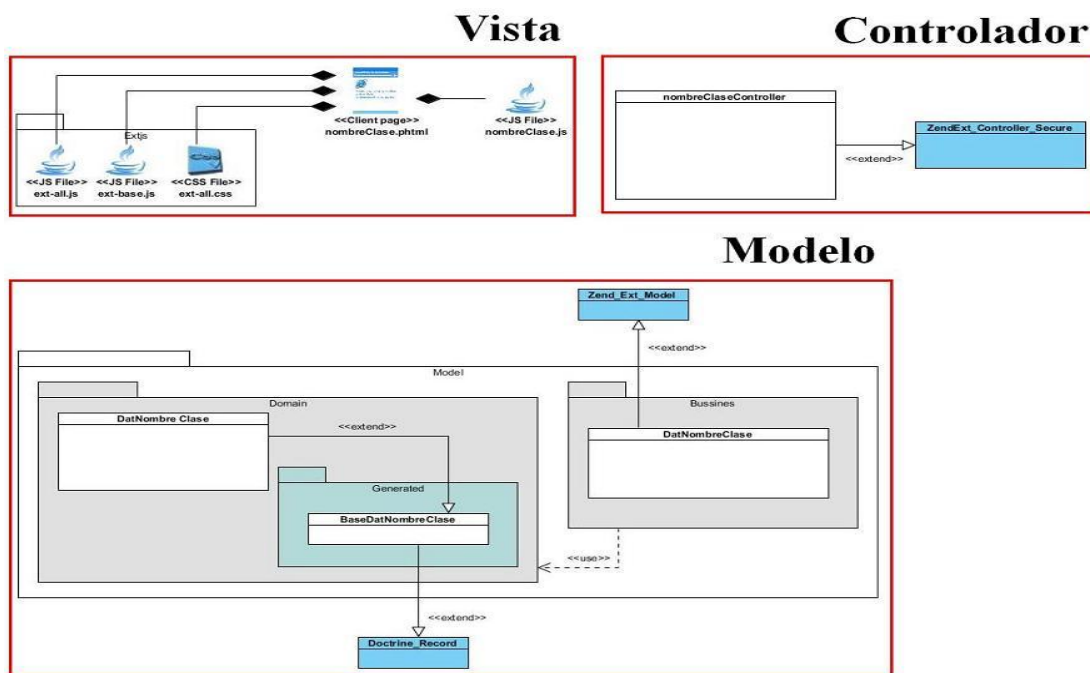


Figura 5: Ejemplo de aplicación del patrón MVC en los diagramas de clase.

3.3 Patrones de Diseño Utilizados

Un patrón de diseño nombra, abstrae e identifica los aspectos clave de un diseño estructurado, común, que lo hace útil para la creación de diseños orientados a objetos reutilizables. Los patrones de diseño identifican las

clases participantes y las instancias, sus papeles y colaboraciones, y la distribución de responsabilidades. Se describe cuando se aplica, las características de otros diseños y las consecuencias y ventajas de su uso (26).

3.3.1 Patrones GOF

Los patrones Gang Of Four (GoF), son patrones de diseño que están divididos fundamentalmente en tres grandes grupos: estructurales, creacionales y de comportamiento.

Como ventaja de la utilización del marco de trabajo Saxe se aplican los patrones que se detallan a continuación:

Fachada: Provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema (26). La utilización de este patrón está reflejada en la creación y empleo de la clase IOC ya que esta funciona como intermediario en la comunicación entre componentes.

Cadena de responsabilidad: Permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada. La cadena de responsabilidad se encarga de evitar el acoplamiento del remitente de una petición a su receptor, dando a más de un objeto la posibilidad de manejar la petición (26). Este patrón se refleja en las clases controladoras y las del modelo donde se establece la una cadena a seguir para las peticiones de la clase controladora al modelo y del modelo a la entidad. Ejemplos: GestionarsituacionpeligrosaController y DatSituacionpeligrosaModel.

Solitario: Es un patrón creacional que garantiza que exista una instancia única para una clase y proporciona un punto de acceso global a ella (27). Se evidencia en las clases del dominio las cuales poseen métodos estáticos a los cuales se puede acceder sin necesidad de crear nuevas instancias de dichas clases.

3.3.2 Patrones GRASP

Los Patrones Generales de Software para Asignación de Responsabilidades (GRASP, por sus siglas en inglés), describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable (28).

Existen nueve patrones GRASP los cuales son: Alta cohesión, Bajo acoplamiento, Controlador, Experto, Creador, Polimorfismo, Fabricación pura, Indirección y No hables con extraños.

A continuación se describen cada uno de los patrones que fueron aplicados en el diseño de la solución:

Experto: Asigna responsabilidad al experto en información: la clase que tiene la información necesaria para la realización de la asignación. Es un patrón que se usa más que cualquier otro al asignar responsabilidades. Conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Soporta un bajo acoplamiento, lo que favorece tener sistemas más robustos y de fácil mantenimiento(27). Este patrón se utilizó en la implementación de varias clases de la aplicación, ejemplo de algunas son las clases controladoras y las clases del modelo, ya que son las que cuentan con la información necesaria para cumplir sus responsabilidades. Ejemplo: `DatSituacionpeligrosaModel`, `DatRiesgolaboralModel` y `GestionarsituacionpeligrosaController`.

Creador: Asigna a la clase B la responsabilidad de crear una instancia de clase A si se cumple uno o más de los casos siguientes:

- B agrega objetos de A

- B contiene objetos de A

- B registra instancias de objetos de A

- B utiliza más estrechamente objetos de A.

B tiene datos de inicialización que se pasarán a un objeto de A cuando sea creado (por tanto, B es un Experto con respecto a la creación de A).

- B es un creador de los objetos A.

Permite crear instancias de otras clases según la responsabilidad de la misma. Esto posibilita un bajo acoplamiento y mejores oportunidades de reutilización(27). Este patrón se utilizó en la implementación de las clases del paquete Domain, quienes son las encargadas de crear los objetos de tipo `Doctrine_Query`, para permitir el acceso a la información almacenada a nivel de datos, ejemplo de algunas son: `DatPlanDeMedidaModel`, `NomMedidaeejecutarModel` y `DatChequeoMedicoModel`.

Controlador: La aplicación del patrón Controlador consiste en asignar la responsabilidad de administrar un mensaje de eventos del sistema a una clase que represente: el negocio, la organización global, o el sistema

global (28). Se utiliza para cada agrupación de requisitos funcionales relacionados los cuales presentan un controlador que atiende a las peticiones de todas sus funcionalidades.

Bajo acoplamiento: Asignar una responsabilidad para mantener bajo acoplamiento. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. Una clase con bajo (o débil) acoplamiento no depende de muchas otras (28). Se emplea en la parte del negocio, ya que solo existen las relaciones necesarias entre las clases para mantener un bajo acoplamiento. Ejemplo: `GestionarsituacionpeligrosaController` y `DatSituacionpeligrosaModel`.

Alta cohesión: Asigna una responsabilidad de modo que la cohesión siga siendo alta. En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. La utilización de este patrón permite que las clases contengan las responsabilidades estrechamente relacionadas, sin tener que realizar un trabajo excesivo, posibilitando la claridad y facilidad con que se entiende el diseño, simplificando el mantenimiento y las mejoras en funcionalidad, logrando en ocasiones un bajo acoplamiento (28). Se emplea en las clases controladoras y del modelo ya que fueron asignadas responsabilidades a las clases de forma tal que la cohesión siguiera siendo alta, o sea, cada clase se encargará de realizar solamente las funciones que estén en correspondencia con la responsabilidad que posea. Ejemplo: `DatRiesgolaboralModel`.

3.4 Diagramas de clases del diseño.

Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases de software, las interfaces y sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema. Son importantes para visualizar, especificar, documentar modelos estructurales y construir sistemas ejecutables aplicando ingeniería directa e inversa(29).

En la Figura 6: **Diagrama de clases del diseño del escenario Gestionar riesgos laborales** se muestra el diagrama de clases del diseño del escenario Gestionar riesgos laborales. Se representan las principales clases, operaciones y relaciones que se necesitan para darle cumplimiento a los requisitos funcionales relacionados con la gestión de los riesgos. Las clases `gestionarRiesgoLaboral.js` y `gestionarRiesgoLaboral.phtml` conforman la capa arquitectónica de presentación. La clase `GestionarriesgolaboralController` solo maneja la

comunicación entre la vista y el modelo, las clases **Model** son las encargadas de la lógica del negocio, implementando funcionalidades que garantizan el cumplimiento de los requisitos identificados; encargadas del acceso a los datos se encuentran **DatRiesgoLaboral**, **NomTiporiesgo** y todas las clases **Base** de las cuales extienden las anteriormente mencionadas. El resto de los diagramas pueden consultarse en los anexos.

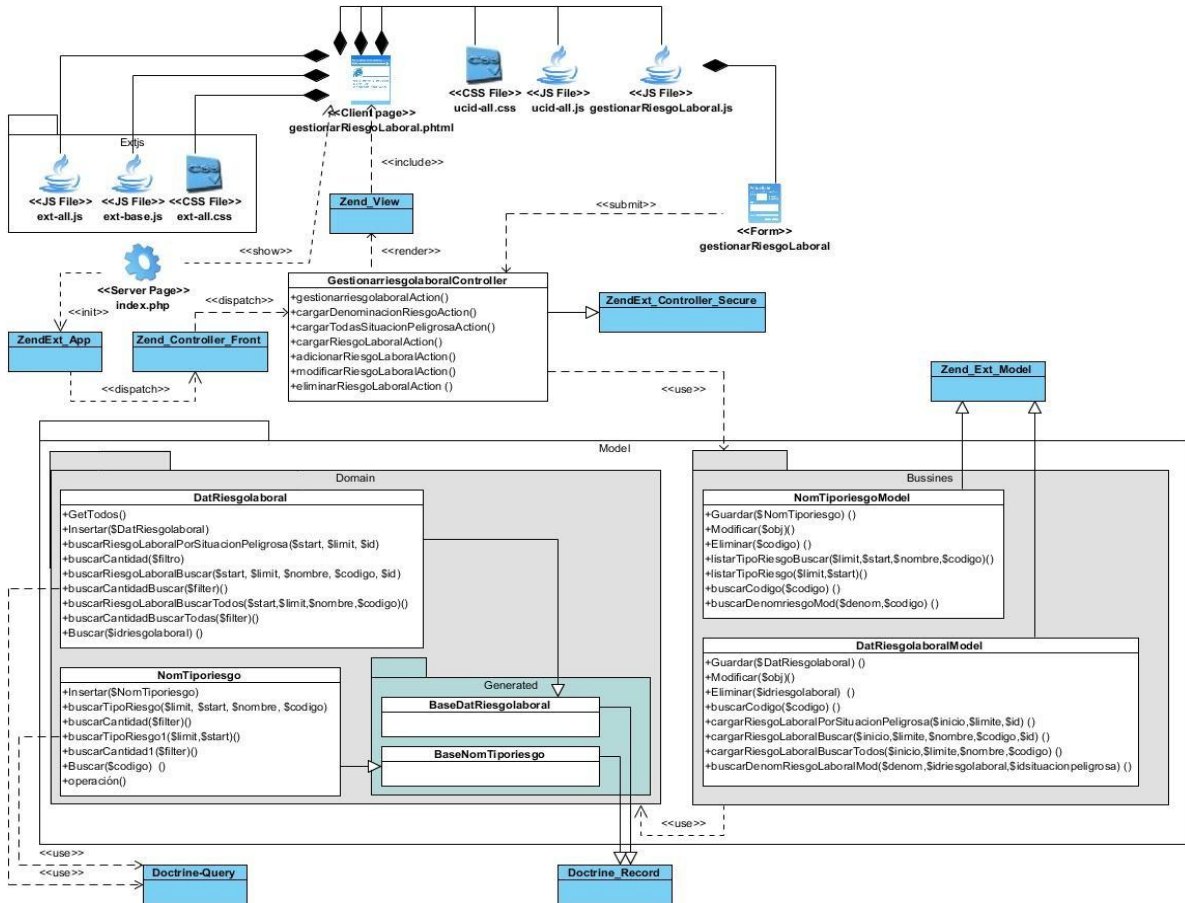


Figura 6: Diagrama de clases del diseño del escenario Gestionar riesgos laborales.

3.5 Diagramas de secuencia.

Los diagramas de secuencia se utilizan para modelar los aspectos dinámicos de un sistema(30). La mayoría de las veces, esto implica modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento.

A continuación se presenta el diagrama de secuencia correspondiente al requisito funcional Adicionar riesgo laboral, el resto de los diagramas se pueden consultar en los anexos.

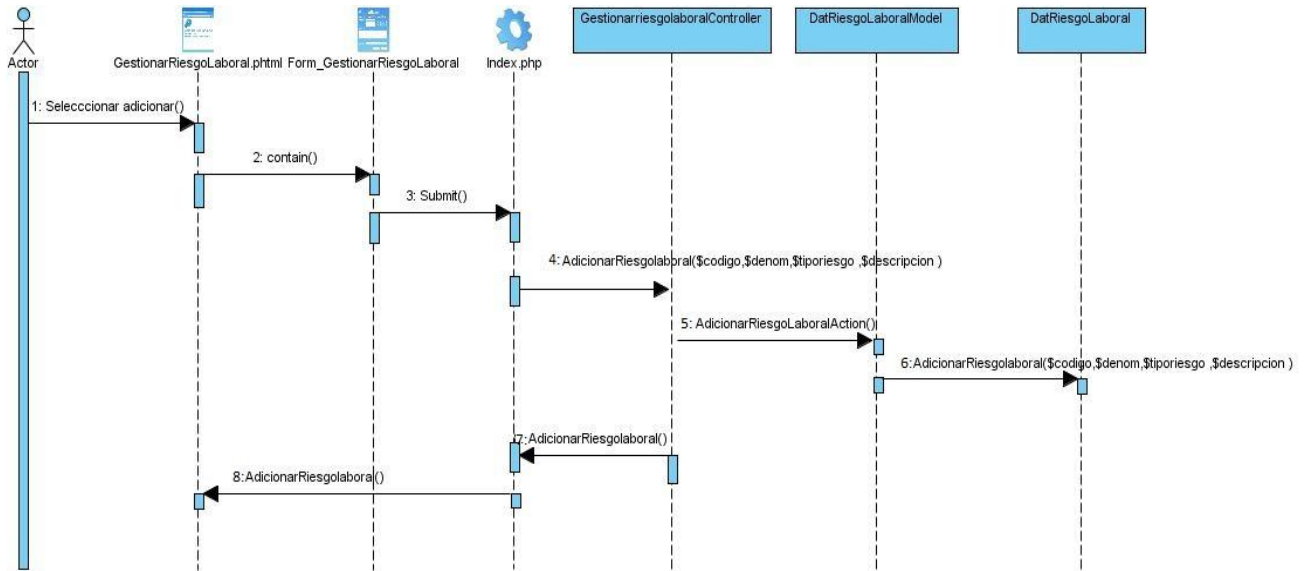


Figura 7: Diagrama de secuencia Adicionar riesgo laboral.

3.6 Modelo de datos

Un modelo de datos es la representación abstracta de los datos en un sistema gestor de base de datos. Básicamente el modelo de datos está formado por tres elementos fundamentales que son: los objetos, que son todas las entidades que manipulan los datos a persistir; los atributos, que son las características básicas de los objetos antes mencionados; y las relaciones, que son las que enlazan a dichos objetos entre sí (30).

El modelo de datos propuesto en la solución cuenta con un total de 24 tablas, de ellas 6 son nomencladores encargados de gestionar conceptos específicos del negocio ya predefinidos, ejemplo: la tabla “nom_tipochequeo” gestiona los tipos de chequeos específicos que se definen en la entidad. De las tablas que se muestran en el modelo de datos, hay 3 que no pertenecen directamente al componente Seguridad y Salud en el Trabajo, pero son imprescindibles para llevar a cabo este proceso, las mismas son la tabla “dat_trabajador”, “dat_puestrotrabajo” y “dat_areatrabajo”.

En la Figura 8: **Modelo de datos del componente Seguridad y Salud en el Trabajo** se muestra el modelo de datos obtenido para el componente Seguridad y Salud en el Trabajo:

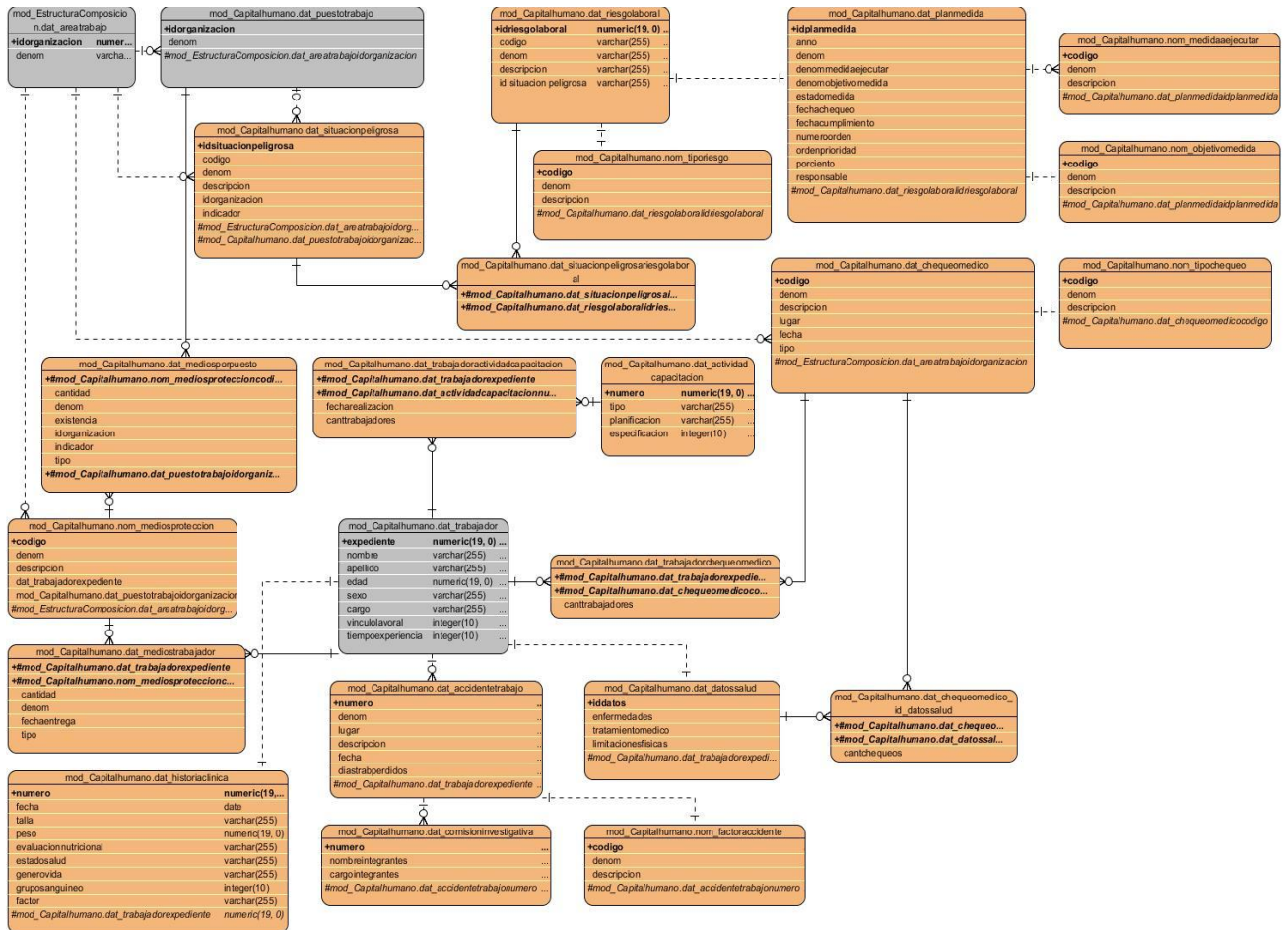


Figura 8: Modelo de datos del componente Seguridad y Salud en el Trabajo

3.7 Diagrama de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema. Un componente representa un módulo de software con una interfaz bien definida (31). El diagrama de componentes correspondiente al componente Seguridad y Salud en el Trabajo del subsistema Capital Humano integrado al sistema CedruX contiene componentes, interfaces, y

relaciones entre ellos, incluyendo paquetes utilizados para agrupar elementos en el modelo. A continuación se muestra el diagrama de componente de la aplicación desarrollada:

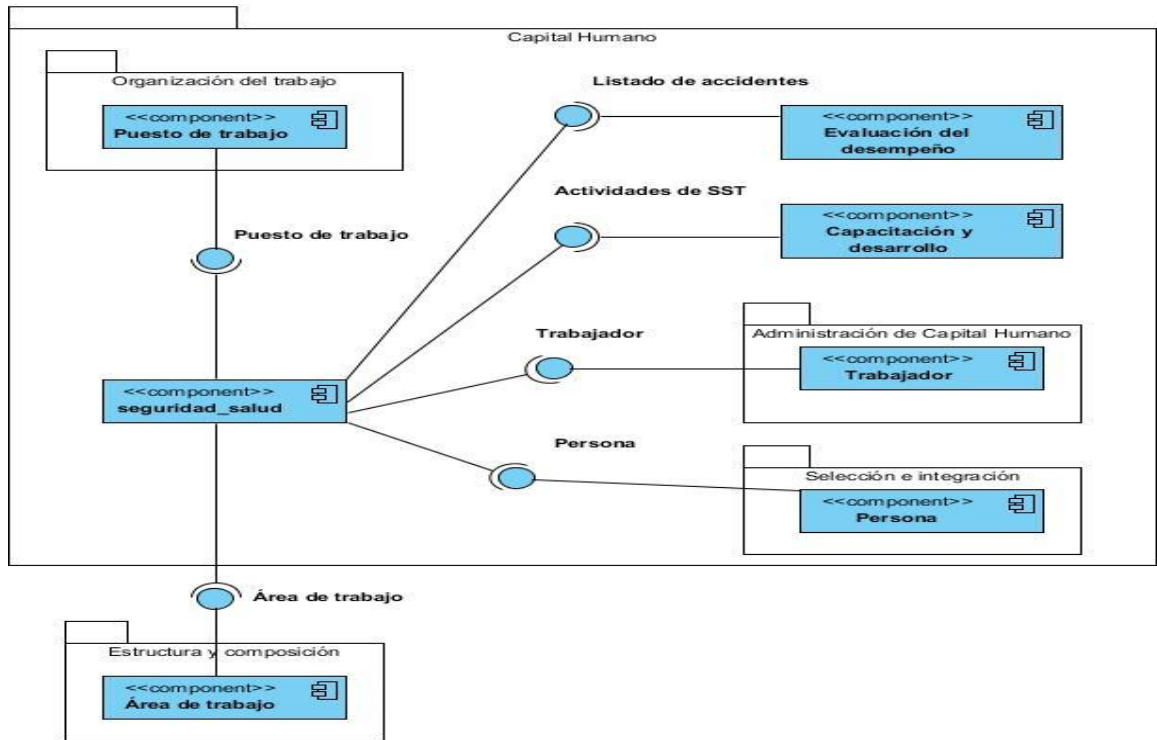


Figura 9: Diagrama de componentes de SST.

3.8 Conclusiones parciales

- La modelación del diseño, permitió identificar las principales clases de la solución, las relaciones existentes entre ellas y sus métodos asociados.
- Se da a conocer el modelo de datos elaborado para el sistema, definiendo la persistencia de los datos en la base de datos.

4. IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción

En el presente capítulo se describe la implementación del sistema, además se hace referencia a las métricas tamaño operacional de clase y relaciones entre clases, las cuales son usadas para la validación del diseño de la propuesta de solución, con el fin de comprobar su correcta realización. Se especifican las pruebas de software que son usadas para la validación y comprobación de la calidad del sistema desarrollado.

4.2 Diagrama de despliegue

El diagrama de despliegue es utilizado para modelar el hardware empleado en las implementaciones de sistemas y las relaciones entre sus componentes. Constituye un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El diagrama de despliegue del componente Seguridad y Salud en el trabajo, se rige por la especificación dada para el subsistema Capital humano. A continuación se muestran los posibles escenarios con los requerimientos de software.

Servidores	Clientes
Servidor de Aplicaciones Web Sistema Operativo: Ubuntu Server Servidor Web: Apache 2.0 Librerías Adicionales: PHP 5	PC Cliente con disco duro Sistema Operativo: Linux o Windows Navegador Web: Mozilla Firefox v2.2 o superior
Servidor de Base de Datos Sistema Operativo: Ubuntu Server Sistema Gestor de Base de Datos: PostgreSQL 8.3.8	PC Cliente sin disco duro Todo se instala en el servidor de clientes ligeros
Servidor de Clientes Ligeros Sistema Operativo: Nova Server Navegador Web: Mozilla Firefox 2.17 o superior Herramienta de administración de clientes ligeros	

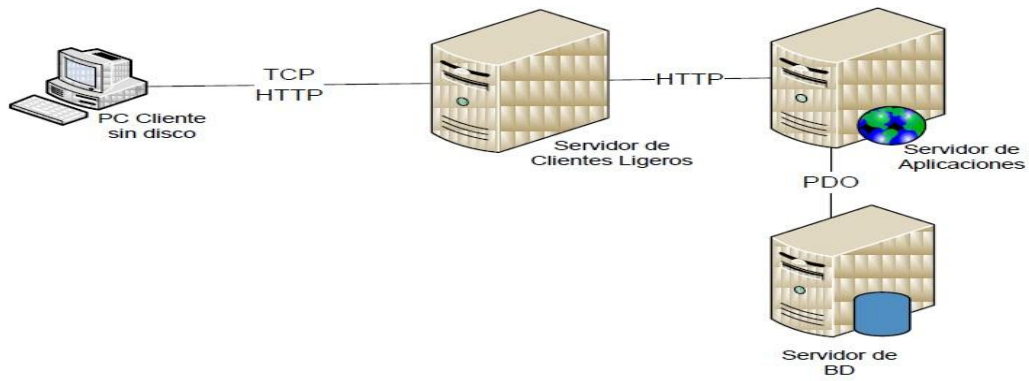


Figura 10: Diagrama de despliegue de escenario para PC cliente sin disco.



Figura 11: Diagrama de despliegue de escenario para PC cliente con disco.

4.3 Funcionalidades implementadas

En la Figura 12 : **Interfaz Gestionar riesgos laborales**. se muestra la interfaz Gestionar riesgos laborales, que ofrece las opciones de Adicionar, Modificar, Eliminar y Buscar riesgos laborales, además permite buscar las situaciones peligrosas y muestra un listado de los riesgos asociados a cada una de ellas, para ver el resto de las interfaces puede consultar los anexos.

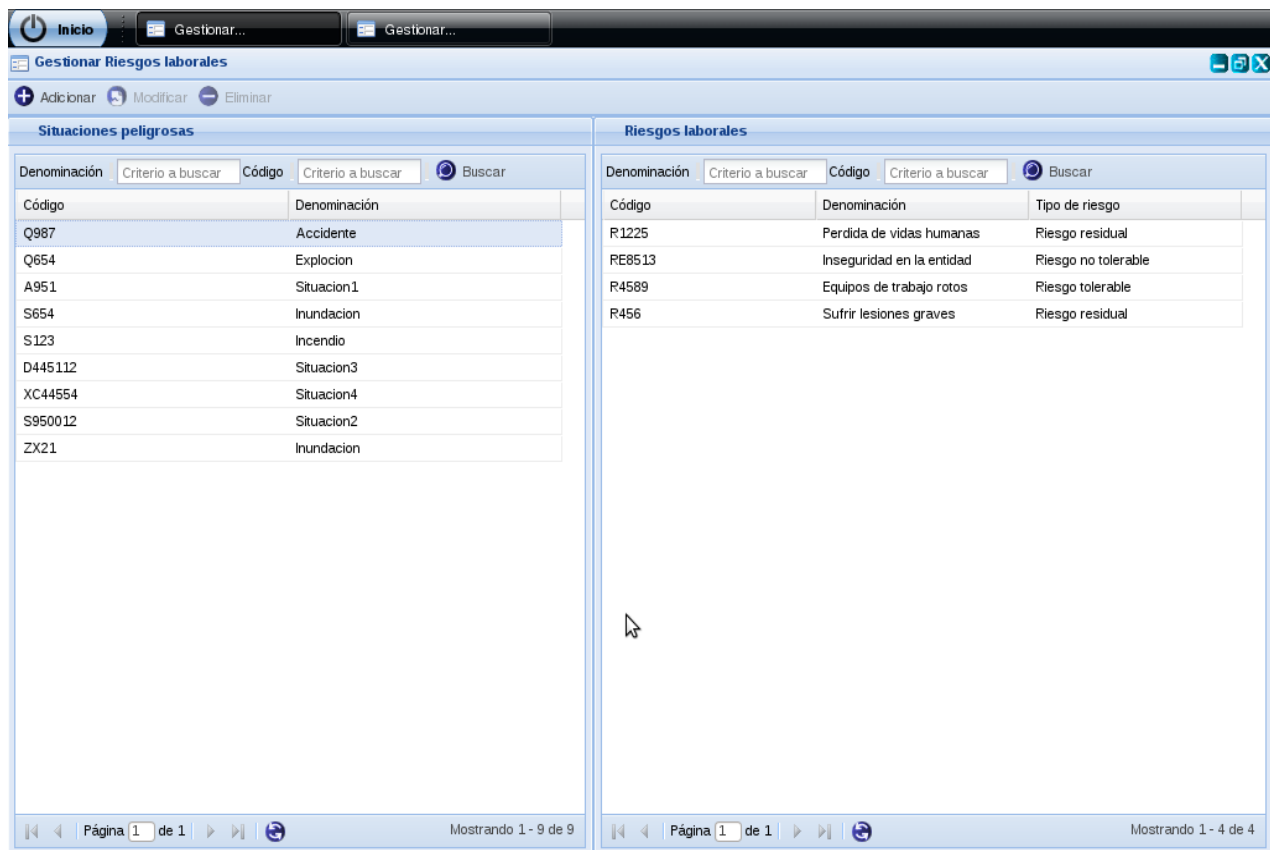


Figura 12 : Interfaz Gestionar riesgos laborales.

4.4 Métricas para la validación del diseño propuesto

Las métricas de software son una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Permiten averiguar cuán bien están definidas las clases y el sistema, lo cual tiene un impacto directo en el mantenimiento del mismo, tanto por la comprensión de lo desarrollado como por la dificultad de modificarlo con éxito, tienen como propósito entender y mejorar la calidad del producto, evaluar la efectividad del proceso y mejorar la calidad del trabajo llevado a cabo al nivel del proyecto (30).

Una vez realizado el diseño del componente Seguridad y Salud en el Trabajo, se llevó a cabo su validación, para esto se utilizaron las métricas orientadas a objetos, específicamente las orientadas a clases.

Las métricas empleadas para evaluar el diseño realizado fueron la métrica Tamaño Operacional de clase (TOC) y la métrica Relaciones entre clases (RC), ya que se ajustan para evaluar al diseño realizado, siendo las mismas de fácil utilización.

Estas métricas están diseñadas para evaluar los siguientes atributos de calidad:

Responsabilidad: Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.

Complejidad de implementación: Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.

Reutilización: Consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.

Acoplamiento: Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, está muy ligada a la característica de Reutilización.

Complejidad de mantenimiento: Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en la planificación del proyecto.

Cantidad de pruebas: Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad del producto (componente, módulo, clase, conjunto de clases) diseñado.

Debido a la cantidad de clases que posee el componente, solo se le aplicó a las clases controladoras y del modelo.

4.4.1 Métrica Tamaño operacional de clase

Tamaño operacional de clase (TOC): Está dado por el número de métodos asignados a una clase. A continuación se muestra una serie de tablas encaminadas a un mejor entendimiento de la utilización de esta métrica.

Atributo que afecta	Modo en que lo afecta
---------------------	-----------------------

Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC implica una disminución del grado de reutilización de la clase.

Tabla 2: Tamaño operacional de clase (TOC).

Atributo	Categoría	Criterio
Responsabilidad	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
Complejidad de implementación	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
Reutilización	Baja	$> 2 \times$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	\leq Promedio

Tabla 3. Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica TOC.

La tabla que se muestra a continuación ofrece una selección de las clases del sistema a las que se le aplicó la métrica y los resultados obtenidos para cada atributo evaluado.

No	Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
1	GestionartiporiesgoController	5	Baja	Baja	Alta
2	GestionartipochequeomedicoController	5	Baja	Baja	Alta
3	GestionarobjetivomedidaController	5	Baja	Baja	Alta
4	GestionarmediosproteccionController	5	Baja	Baja	Alta
5	GestionarmedidaaejecutarController	5	Baja	Baja	Alta
6	GestionarfactoraccidenteController	5	Baja	Baja	Alta
7	NomTiporiesgoModel	7	Media	Media	Alta
8	NomTipochequeoModel	7	Media	Media	Alta
9	NomObjetivomedidaModel	7	Media	Media	Alta
10	NomMediosproteccionModel	7	Media	Media	Alta
11	NomMedidaaejecutarModel	7	Media	Media	Alta
12	NomFactoraccidenteModel	7	Media	Media	Alta
13	NomTiporiesgo	6	Baja	Baja	Alta

14	NomTipochequeo	6	Baja	Baja	Alta
15	NomObjetivomedida	6	Baja	Baja	Alta
16	NomMediosproteccion	6	Baja	Baja	Alta
17	NomMedidaaejecutar	6	Baja	Baja	Alta
18	NomFactoraccidente	6	Baja	Baja	Alta
19	GestionarsituacionpeligrosaControl ntroller	6	Baja	Baja	Alta
20	GestionarriesgolaboralControl er	6	Baja	Baja	Alta
21	GestionarplandemedidaControl ller	11	Media	Media	Media
22	DatSituacionpeligrosaModel	9	Media	Media	Media
23	DatRiesgolaboralModel	9	Media	Media	Media
24	DatPlanmedidaModel	10	Media	Media	Media
25	DatSituacionpeligrosa	10	Media	Media	Media

Tabla 4: Resultados de la aplicación de la métrica TOC para cada clase del sistema.

Las gráficas que se muestran a continuación muestran los resultados obtenidos para cada uno de los atributos medidos.

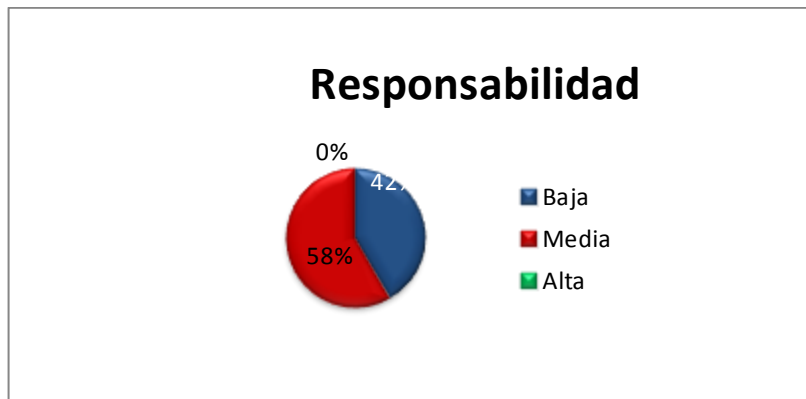


Figura 13: Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.

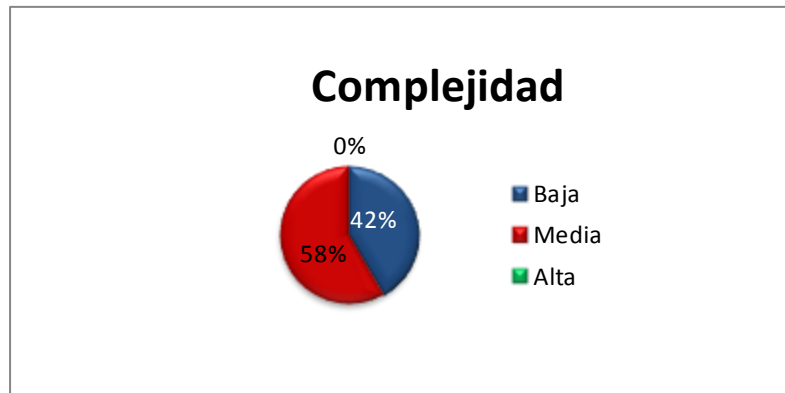


Figura 14: Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación.

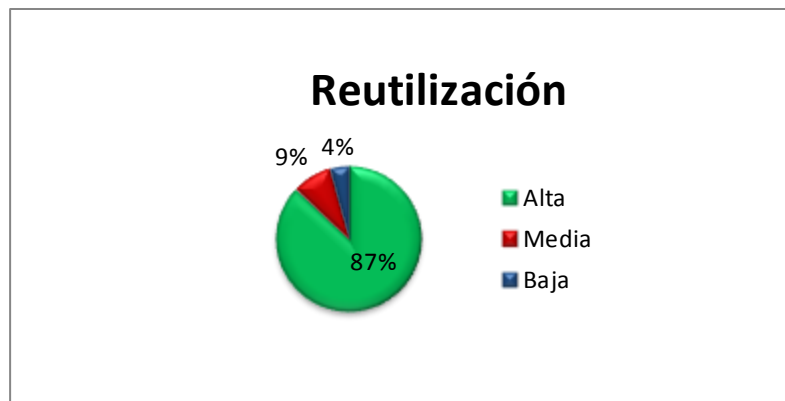


Figura 15: Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.

Al realizar un análisis de los resultados obtenidos al aplicarle esta métrica a las clases controladoras y a las del modelo se demuestra que la mayoría de estas clases se encuentran dentro de las categorías media y baja, quedando demostrado de esa forma la elevada reutilización, baja complejidad y responsabilidad que presenta el diseño propuesto, permitiendo una sencilla implementación y una amplia realización de pruebas con mayor facilidad, por lo que los resultados obtenidos según esta métrica son positivos.

4.4.2 Métrica Relaciones entre clases

Relaciones entre clases (RC): Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad.

Atributo que afecta	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del acoplamiento de la clase.
Complejidad de mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Un aumento del RC implica una disminución en el grado de reutilización de la clase.

Cantidad de pruebas	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.
---------------------	--

Tabla 5: Relaciones entre clases (RC).

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
Complejidad de mantenimiento	Baja	\leq Promedio
	Media	Entre Promedio y $2 \cdot$ Promedio
	Alta	$>2 \cdot$ Promedio
Reutilización	Baja	$>2 \cdot$ Promedio
	Media	Entre Promedio y $2 \cdot$ Promedio
	Alta	\leq Promedio
Cantidad de pruebas	Baja	\leq Promedio
	Media	Entre Promedio y $2 \cdot$ Promedio
	Alta	$>2 \cdot$ Promedio

Tabla 6: Rango de valores para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas) relacionados con la métrica RC.

La tabla 7 ofrece una selección de las clases del sistema a las que se le aplicó la métrica y los resultados obtenidos para cada atributo evaluado.

No	Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad Mant	Reutilización	Cantidad de Pruebas
1	GestionartiporiesgoController	1	Bajo	Media	Media	Media
2	GestionartipochequeomedicoController	1	Bajo	Media	Media	Media
3	GestionarobjetivomedidaController	1	Bajo	Media	Media	Media
4	GestionarmediosproteccionController	1	Bajo	Media	Media	Media
5	GestionarmedidaaejecutarController	1	Bajo	Media	Media	Media
6	GestionarfactoraccidenteController	1	Bajo	Media	Media	Media
7	NomTiporiesgoModel	1	Bajo	Media	Media	Media
8	NomTipochequeoModel	1	Bajo	Media	Media	Media
9	NomObjetivomedidaModel	1	Bajo	Media	Media	Media
10	NomMediosproteccionModel	1	Bajo	Media	Media	Media

11	NomMedidaaexecutarModel	1	Bajo	Media	Media	Media
12	NomFactoraccidenteModel	1	Bajo	Media	Media	Media
13	NomTiporiesgo	0	Ninguno	Baja	Alta	Baja
14	NomTipochequeo	0	Ninguno	Baja	Alta	Baja
15	NomObjetivomedita	0	Ninguno	Baja	Alta	Baja
16	NomMediosproteccion	0	Ninguno	Baja	Alta	Baja
17	NomMedidaaexecutar	0	Ninguno	Baja	Alta	Baja
18	NomFactoraccidente	0	Ninguno	Baja	Alta	Baja
19	GestionarsituacionpeligrosaController	2	Medio	Alta	Baja	Alta
20	GestionarriesgolaboralController	3	Alto	Alta	Baja	Alta
21	GestionarplandemeditaController	3	Alto	Alta	Baja	Alta
22	DatSituacionpeligrosaModel	1	Bajo	Media	Media	Media
23	DatRiesgolaboralModel	1	Bajo	Media	Media	Media
24	DatPlanmedidaModel	1	Bajo	Media	Media	Media

25	DatSituacionpeligrosa	0	Ninguno	Baja	Alta	Baja
----	-----------------------	---	---------	------	------	------

Tabla 7: Resultados de la aplicación de la métrica RC para cada clase del sistema.

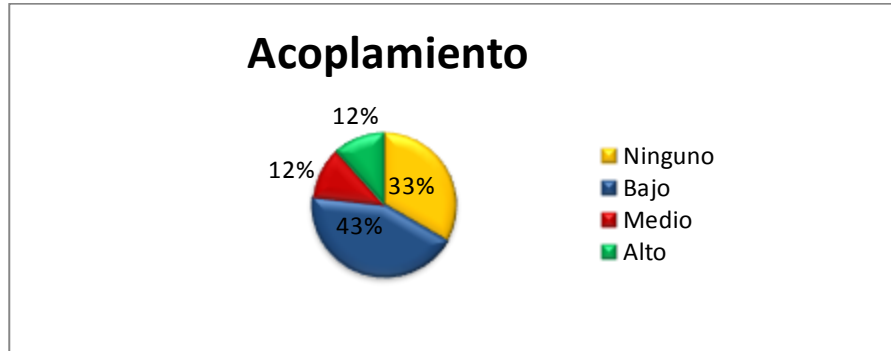


Figura 16: Representación en % de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.

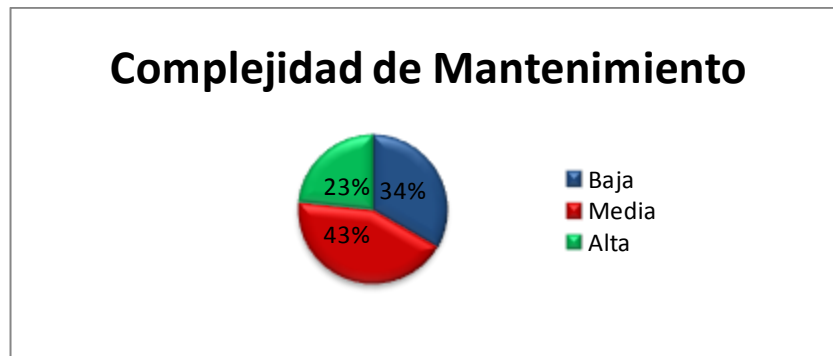


Figura 17: Representación en % de los resultados de la evaluación de la métrica RC en el atributo Complejidad de mantenimiento.

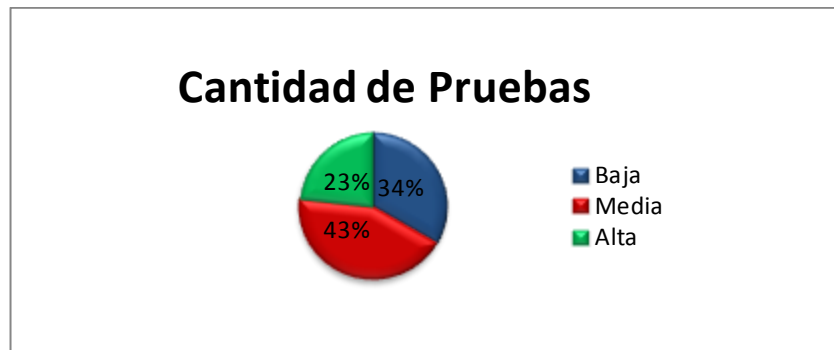


Figura 18: Representación en % de los resultados de la evaluación de la métrica RC en el atributo Reutilización.



Figura 19: Representación en % de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas.

Luego de aplicar la métrica de diseño RC y obtenidos los resultados de la evaluación del instrumento de medición de la métrica, se puede concluir que el diseño propuesto tiene una calidad aceptable teniendo en cuenta que la mayoría de las clases empleadas poseen menos de 3 dependencias de otras clases, lo que lleva a evaluaciones positivas de los atributos de calidad involucrados (acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización). Favoreciendo de esta manera la reutilización de las clases así como la modificación e implantación del diseño.

4.5 Pruebas de software

Las pruebas de software son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es

hecha de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación (32).

4.5.1 Pruebas de caja blanca

La prueba de caja blanca, o pruebas de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Para la solución desarrollada la prueba de Caja Blanca aplicada fue la del camino básico.

Esta técnica de prueba permite obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa (32).

Para realizar la prueba del Camino básico es preciso calcular la complejidad ciclomática del algoritmo o fragmento de código a analizar. A continuación se muestra el código del método vincularTrabajadoresActAction encargado de vincular trabajadores una actividad.

```
function vincularTrabajadoresActAction()
{
  extract($_REQUEST);1
  $$trabajadoresArr = json_decode(stripslashes($_datos));1
  $trabajadoractividadmodel = new DatTrabajadoractividadcapacitacionModel();1
  $cont=0;1
  for($j =0; $j< count($$trabajadoresArr);$j++){2
    $existe = $trabajadoractividadmodel->buscarTrabajador($trabajadoresArr[$j]->idtrabajador,$trabajadoresArr[$j]->idactividad);3
    if($existe!=1){4
      $obj = new DatTrabajadoractividadcapacitacion();5
      $obj->nombre = $trabajadoresArr[$j]->nombre;5
      $obj->papel = $trabajadoresArr[$j]->papel;5
      $obj->sapel = $trabajadoresArr[$j]->sapel;5
      $obj->idtrabajador = $trabajadoresArr[$j]->idtrabajador;5
      $obj->numexpediente = $trabajadoresArr[$j]->numexpediente;5
      $obj->puestotrabajo = $trabajadoresArr[$j]->puestotrabajo;5
      $obj->area = $trabajadoresArr[$j]->area; 5
      $obj->idactividad = $trabajadoresArr[$j]->idactividad;5
      $result = $trabajadoractividadmodel->Guardar($obj);5
      $cont++;5
    }6
  }7
  echo $cont;return;8
}
```

Figura 20: Código del método vincularTrabajadoresActAction.

Para el cálculo de la complejidad ciclomática es necesario representar el grafo de flujo asociado al código antes presentado a través de nodos, aristas y regiones, quedando como se muestra en la figura.

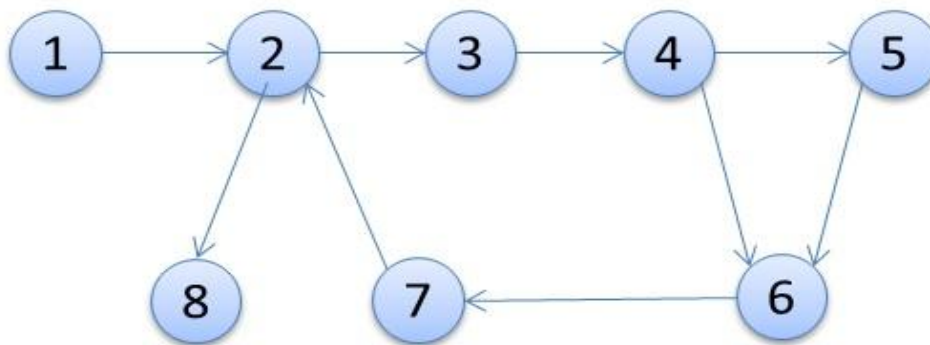


Figura 21: Grafo de flujo asociado al método.

Una vez construido el grafo de flujo asociado al procedimiento anterior se determina la complejidad ciclomática, el cálculo es necesario efectuarlo mediante tres vías o fórmulas, se debe utilizar el mismo grafo en cada caso:

Cálculo de la complejidad ciclomática a partir de un segmento de código.

La complejidad ciclomática es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto del método de prueba del camino básico, el valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez (32).

La complejidad ciclomática de un código se puede calcular de tres maneras diferentes.

Para calcular la complejidad ciclomática del método **vincularTrabajadoresActAction ()** fueron utilizadas estas tres formas para lograr una amplia verificación de los resultados.

Las fórmulas para realizar dicho cálculo son:

$$1. \quad V(G) = (A - N) + 2$$

$$V(G) = (9 - 8) + 2$$

$$V(G) = 3$$

Siendo A la cantidad total de aristas del grafo y N la cantidad de nodos.

$$2. \quad V(G) = P + 1$$

$$V(G) = 2 + 1$$

$$V(G) = 3$$

Siendo P la cantidad de nodos predicado (son aquellos de los cuales parten dos o más aristas)

$$3. \quad V(G) = R$$

$$V(G) = 3$$

Siendo R la cantidad de regiones que posee el grafo.

En cada una de las fórmulas V (G) representa el valor del cálculo. Según los resultados obtenidos en cada uno de estos cálculos se puede concluir que la complejidad ciclomática del código analizado es 3, determinándose a su vez que existen tres caminos posibles por donde puede circular el flujo y que esta misma cantidad representa el límite superior de casos de prueba que se le pueden aplicar a dicho código.

A continuación se muestran los caminos básicos por donde puede circular el flujo.

Camino básico # 1: 1-2-8

Camino básico # 2: 1-2-3-4-6-7-2-8

Camino básico # 3: 1-2-3-4-5-6-7-2-8

Para cada uno de los caminos obtenidos se realiza un caso de prueba. Los casos de prueba realizados son los siguientes:

Caso de prueba para el camino básico #1.

Camino básico # 1: 1-2-8

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro \$trabajadoresArr está vacío, no contiene los datos necesarios para vincularlos trabajadores a la actividad de capacitación, por lo que no se puede realizar la operación de Vincular trabajadores.

Entrada: \$trabajadoresArr= [\$trabajadoresArr [0] ->idtrabajador=>0, \$trabajadoresArr [0] ->idactividad=> 0].

Resultados esperados: Se lanza el mensaje “Debe seleccionar al menos un trabajador”.

Caso de prueba para el camino básico #2.

Camino básico # 2: 1-2-3-4-6-7-2-8

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro \$trabajadoresArr no está vacío, contiene todos los datos del trabajador y de la actividad de capacitación, necesarios para realizar la operación de Vincular trabajadores.

Entrada: \$trabajadoresArr = [\$trabajadoresArr [0] ->idtrabajador => 900001238, \$trabajadoresArr [0] ->idactividad => 1, \$trabajadoresArr [1] ->idtrabajador => 900001240, \$trabajadoresArr [1] ->idactividad => 1].

Resultados esperados: La actividad de capacitación tiene vinculado al menos un trabajador de los seleccionados previamente, se espera que no sea asociado el trabajador a la actividad de capacitación.

Caso de prueba para el camino básico #3.

Camino básico # 3:1-2-3-4-5-6-7-2-8

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro \$ trabajadoresArr no está vacío, contiene todos los datos del trabajador y de la actividad de capacitación, necesarios para realizar la operación de Vincular trabajadores.

Entrada: \$trabajadoresArr = [\$trabajadoresArr [0] ->idtrabajador => 900001238,

\$trabajadoresArr [0] ->idactividad => 1, \$trabajadoresArr [1] ->idtrabajador => 900001240,

\$trabajadoresArr [1] ->idactividad => 1].

Resultados esperados: La actividad de capacitación no tiene vinculado el o los trabajadores seleccionados, se espera que sean asociados los trabajadores a la actividad de capacitación.

Resultado de las pruebas de Caja blanca

Para la realización de las pruebas de caja blanca al sistema, fueron analizadas 15 funcionalidades significativas para el componente, lo que permitió detectar errores en la implementación de 8 de ellas. Luego de corregir los errores encontrados, se pudo comprobar que el flujo de trabajo de las funciones está correcto ya que cumple con las condiciones necesarias que se habían planteado anteriormente.

4.5.2 Pruebas de caja negra

Las pruebas de caja negra son las que se aplican a la interfaz del software. Una prueba de este tipo examina algún aspecto funcional de un sistema que tiene poca relación con la estructura lógica interna del software. Las pruebas de caja negra, también denominadas pruebas de comportamiento, se concentran en los requisitos funcionales del software, es decir permiten al ingeniero de software derivar conjuntos de condiciones de entrada que ejercerán por completo todos los requisitos funcionales de un programa. Tratan de encontrar errores de las siguientes categorías: funciones incorrectas o faltantes, errores de interfaz, errores de estructuras de datos o en acceso a datos externos, errores de comportamiento o desempeño y errores de inicialización y término (32).

Existen diferentes técnicas de prueba de Caja negra descritas para validar la funcionalidad del sistema sin entrar a analizar su ejecución interna.

- Métodos de prueba basados en grafos.
- Análisis de valores límites.
- Tabla ortogonal.
- Partición de equivalencia.

De estas técnicas, la seleccionada fue partición de equivalencia la cual permite examinar los valores válidos e inválidos de las entradas existentes en el software.

Para la aplicación de esta técnica se realizan los diseños de casos de prueba los cuales se basan en una evaluación de las clases de equivalencia para una condición de entrada.

Para validar el componente atendiendo a la técnica partición de equivalencia Se realizaron 110 diseños de casos de prueba. A continuación se especifica el caso de prueba para el requisito “Adicionar riesgo laboral”.

Condiciones de ejecución

- Se debe identificar y autenticar ante el sistema y además debe tener los permisos para ejecutar esta acción.
- Se debe pulsar el botón Adicionar de la barra de tareas de la ventana Gestionar Riesgos laborales.
- Debe haber registrado en el sistema al menos una situación peligrosa.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Adicionar riesgo laboral.	Se adiciona un EP riesgo laboral con los atributos: válidos Código, Denominación, Descripción y Tipo	1.1: Adicionar riesgo laboral introduciendo datos presionando el botón Aceptar.	<ul style="list-style-type: none"> ➤ Se selecciona la situación peligrosa. ➤ Se presiona el botón Adicionar. ➤ Se introducen los datos del riesgo laboral correctamente.

de riesgo.

- Se presiona el botón Aceptar.
- Se guardan los datos, se cierra automáticamente la ventana de Adicionar riesgo laboral y se muestra un mensaje de información.
- Se presiona el botón Aceptar.

EP 1.2: Adicionar riesgo laboral introduciendo datos válidos presionando el botón Aplicar.

- Se selecciona la situación peligrosa.
- Se presiona el botón Adicionar.
- Se introducen los datos del riesgo laboral correctamente.
- Se presiona el botón Aplicar.
- Se adiciona un nuevo riesgo laboral y se muestra un mensaje de información.
- Se presiona el botón Aceptar.
- Se limpian los campos de la ventana Adicionar riesgo laboral y ésta permanece abierta.

EP1.3: Adicionar riesgo laboral dejando campos requeridos en blanco.

- Se selecciona la situación peligrosa.
 - Se presiona el botón Adicionar.
 - Se introducen los datos dejando campos requeridos en blanco.
-

	<ul style="list-style-type: none"> ➤ Se presiona el botón Aceptar. ➤ Se presiona el botón Aceptar del mensaje de error.
EP 1.4: Adicionar riesgo laboral introduciendo errores en los datos.	<ul style="list-style-type: none"> ➤ Se selecciona la situación peligrosa. ➤ Se presiona el botón Adicionar. ➤ Se introducen los datos del riesgo laboral que se desea adicionar en el formulario introduciendo errores en los mismos. ➤ Se presiona el botón Aceptar. ➤ Se presiona el botón Aceptar del mensaje de error.
EP1.5: Cancelar	<ul style="list-style-type: none"> – Se selecciona la situación peligrosa. – Se presiona el botón Adicionar. – Se introducen o no los datos en el formulario. – Se presiona el botón Cancelar.

Descripción de variable

No	Nombre del Tipo campo	Válido	Inválido	Inválido
----	-----------------------	--------	----------	----------

laboral humanas) primer muestra el siguiente mensaje:
 introduciend riesgo “Se ha adicionado el riesgo
 o datos laboral) laboral satisfactoriamente.”
 válidos
 presionando
 el botón
 Aplicar.

EP1.3	Adicionar riesgo laboral dejando campos requeridos en blanco.	I(Vacío) V(RL1)	V(Pérdid a de vidas humanas) I(Vacío)	V(Este es primer riesgo laboral) V(Este es primer riesgo laboral)	V(Riesgo el residual) V(Riesgo el residual)	El sistema muestra el siguiente mensaje: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)”. Se muestra marcado en color rojo el campo requerido.	NA
		V(RL1)	V(Pérdid a de vidas humanas)	I(Vacío)	V(Riesgo residual)		
		V(RL1)	V(Pérdid a de vidas humanas)	V(Este es primer riesgo laboral)	I(Vacío)		

EP1.4	Adicionar riesgo laboral introduciendo errores en los datos.	I(D<>/[])	V(Pérdida de vidas humanas)	V(Este es el primer riesgo laboral)	NA	El sistema muestra el siguiente mensaje: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).” Se muestra marcado en color rojo el campo requerido.	NA
		V(RL1)	I(Pérdida 12*>)	V(Este es el primer riesgo laboral)	NA		
		V(RL1)	V(Derrumbe)	I(Este es el primer riesgo laboral 12/*)	NA		
EP1.5	Cancelar	NA				Se cancela la operación y se cierra la ventana.	NA

Resultado de las pruebas de Caja negra

Las pruebas de caja negra a la solución fueron desarrolladas por el Grupo de Aseguramiento de la Calidad del Centro CEIGE. Estas se realizaron en tres iteraciones, donde finalmente se comprobó en la tercera iteración que el componente estaba libre de no conformidades, culminando así la fase de pruebas internas. En los anexos se puede observar el acta de liberación emitida.

La tabla que se muestra a continuación ofrece detalles de los resultados obtenidos durante la fase de prueba de caja negra, la misma representa la cantidad de no conformidades detectadas para cada iteración. Se clasifican en significativas y no significativas según su complejidad y grado de impacto en la solución.

Iteraciones	Cantidad de NC	Tipo		Significativas	No significativas
		Documentación	Aplicación		
1ra	38	20	18	15	23
2da	12	2	10	8	4
3ra	0	0	0	0	0

4.6 Conclusiones parciales

- Se implementaron todas las clases diseñadas logrando un módulo de seguridad y salud en el trabajo funcional y que se adecua a las normas de Cedrux.
- Se exponen las métricas utilizadas para la validación del diseño de la propuesta de solución, las que generaron como resultado que el diseño realizado era simple y con una calidad aceptable.
- Se describieron y aplicaron las pruebas de caja blanca y caja negra, para evaluar el sistema desarrollado, las cuales mostraron que el sistema contaba con un adecuado funcionamiento, demostrando así el cumplimiento de las necesidades del cliente y la calidad requerida en el mismo.

CONCLUSIONES

- Se elaboró el marco teórico de la investigación donde se realizó un estudio de varios sistemas de gestión que incluyen el proceso de seguridad y salud en el trabajo analizando las ventajas y desventajas de cada uno, lo cual evidenció que era más factible crear una solución propia que respondiera a las particularidades de la economía cubana y cumpliera con el paradigma de independencia tecnológica por el que aboga el país.
- Se realizó el diseño de la solución propuesta donde se generaron varios artefactos a partir de la captura y especificación de los requisitos, lo cual permitió obtener los diagramas de clase para cada escenario, los diseños de casos de prueba por cada requisito para un total de 110 diseños y el modelo de datos con un total de 24 tablas para dar paso a la implementación del componente.
- Se desarrolló el componente Seguridad y Salud en el Trabajo del Cedrux a partir del diseño propuesto, el cual permite la gestión de este proceso como establecen las normas vigentes en el país para las entidades empresariales presupuestadas.
- Se realizó la validación del diseño aplicando métricas donde los resultados arrojados demuestran la presencia de valores positivos en los indicadores de calidad medidos.
- Se realizaron las pruebas de caja blanca y caja negra con el objetivo de minimizar la ocurrencia de errores del sistema antes de su implantación en las entidades nacionales, las cuales mostraron que el sistema contaba con un adecuado funcionamiento, demostrando así el cumplimiento de las necesidades del cliente y la calidad requerida en el mismo.

RECOMENDACIONES

- Agregar al componente seguridad y salud en el trabajo funcionalidades que permitan notificar la realización de los chequeos médicos a los trabajadores, utilizando el componente Notificaciones que provee el Marco de trabajo Sauxe.
- Adicionar en el modelo de datos del componente Seguridad y Salud en el Trabajo, una nueva tabla donde se registre la asociación de los accidentes de trabajo que históricamente han ocurrido por áreas, que permita identificar los accidentes más comunes mediante la obtención de un reporte.
- Profundizar en temas referentes a la gestión de seguridad y salud en el trabajo para la implementación de futuras versiones del componente.
- Realizar el despliegue del componente propuesto como parte del subsistema de Capital Humano del sistema Cedrux.

REFERENCIAS BIBLIOGRÁFICAS

1. G. Martínez, Larissa .Administración de recursos humanos. 2004. págs. 8-9.
 2. Dra. Silvia Gil Fundora, Dr. Wilfredo Francisco Martín.Implementación del Sistema de Dirección y Gestión Empresarial: Sistema de Seguridad y Salud en el Trabajo. 2011.
 3. Padilla, Lillian.PLANEACION DE LOS RECURSOS DE LA EMPRESA: ERP. 2007.
 4. PROCEDIMIENTO PARA EL DIAGNÓSTICO DE SEGURIDAD Y SALUD DEL TRABAJO (SST). Hidelvys Cantero Cora, Elisa Leyva Cardeñosa , Ricardo Rojas Casas , Tania Ballester Marsal. 2012.
 5. Isasi, Alina de la Concepción, Roblejo, Yenlis González y Guibert, Pedro Valdés. Estudio sobre la información que se gestiona de los recursos humanos, en los ERP. 2010.
- 6RESOLUCION No. 19/03
7. Morales, Aylín, Caballero, Juan Alberto y Abrante, Yunet Suárez.COMPONENTE DE SOFTWARE PARA GESTIONAR COMPETENCIAS LABORALES MEDIANTE CEDRUX. 2011.
 8. CEIGE.CEIGE-Modelo de Desarrollo de Software. La Habana : s.n., 2012.
 9. Borbón, Yoandry Morejón, Baryolo, Oiner Gómez y García, Darien. ARQUITECTURA TECNOLÓGICA PARA EL DESARROLLO DE SOFTWARE.
 10. Pérez, Mario Raúl. [En línea] 2011. [Citado el: 23 de 11 de 2012.] <https://sites.google.com/site/ZendFrameworkExtJsDoctrine/guias/doctrine..>
 11. Souza, Ana Carolina Carvalcho de Paula. EspWeb. [En línea] 2011. [Citado el: 23 de 11 de 2012.] <http://www.espweb.uem.br/wp/wp-content/uploads/2011/09/anacarolina.pdf>.
 12. Hernandis, José Alberto.<http://www.versionero.com>. [En línea] 2009. [Citado el: 23 de 11 de 2013.] <http://www.versionero.com/noticia/210/visualparadigm->.
 13. Ubuntu, Guía. GUÍA DOCUMENTADA PARA UBUNTU. Subversion. [En línea] 2009. [Citado el: 23 de 11 de 2012.] <http://www.guia-ubuntu.org/index.php?title=Subversion>.
 14. Valdés, Damián Pérez. <http://www.maestrosdelweb.com>. ¿Qué es Javascript? [En línea] 2007. <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
 15. Chávez, Joel Manrique. Lenguaje de Programación PHP. www.monografias.com. [En línea] 4 de 11 de 2006. [Citado el: 27 de Noviembre de 2012.] <http://www.monografias.com/trabajos38/programacion-php/programacion-php.shtml>.
 16. Ibarra, Antonio Aliaga.www.iessanvicente.com. [En línea] 21 de Enero de 2008. [Citado el: 27 de Noviembre de 2012.] <http://www.iessanvicente.com/colaboraciones/postgreSQL.pdf>.

17. Foundation, The Apache Software. www.apache.org. [En línea] 2008. [Citado el: 27 de Noviembre de 2012.] <http://projects.apache.org/indexes/quick.html>.
18. López, Alejandro Cadavid. Mozilla Firefox, el navegador web del momento. [En línea] 2009. [Citado el: 27 de Noviembre de 2012.] <http://www.maestrosdelweb.com/editorial/firefox/>.
19. CAMILO ESTEBAN MAHECHA BADILLO, RAUL ORTIZ. www.calidadindustriaalimentaria.wordpress.com/. www.calidadindustriaalimentaria.wordpress.com/2009/07/14/mapa-de-procesos/. [En línea] 14 de julio de 2009. [Citado el: 6 de marzo de 2013.]
20. Ponce, Ma. Eugenia López. Definición y Comprensión del Proceso.
21. www.buenastareas.com. www.buenastareas.com/ensayos/Modelos-Conceptuales/1979709.html. [En línea] [Citado el: 4 de marzo de 2013.]
22. Nora Koch, María José Escalona. Ingeniería de Requisitos en Aplicaciones para la Web. Sevilla : s.n., 2002.
23. <http://elvex.ugr.es>. <http://elvex.ugr.es/idbis/db/docs/design/2-requirements.pdf>. [En línea] [Citado el: 6 de 3 de 2013.]
24. Laguna, Miguel A. www.infor.uva.es. www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf. [En línea] [Citado el: 10 de marzo de 2013.]
25. Ochoa, Prof. Sergio. Diseño y Arquitectura. 2005.
26. JUAN, FRANCISCO JAVIER MARTÍNEZ. GUÍA DE CONSTRUCCIÓN DE SOFTWARE EN JAVA CON PATRONES DE DISEÑO .
27. Schmidt, Douglas. PATTERN-ORIENTED SOFTWARE ARCHITECTURE VOLUME 2: Patterns for Concurrent and Networked Objects. 2000.
28. ¿Qué es un Patrón de Diseño? Tedeschi, Nicolás. 2006.
29. Marcello Visconti, Hernán Astudillo. Fundamentos de Ingeniería de Software.
30. Pressman, Roger S. Ingeniería del Software: Un enfoque práctico. 2005.
31. Rearte, Ing. Emilio. Base de Datos y UML. 2002.
32. Pressman. Ingeniería del Software: Un enfoque práctico. 2005.
33. Ana Celia Botta Parapar, Guadalupe Martínez Suárez. Rediseño del Subsistema de Relaciones Financieras del sistema de Perfeccionamiento Empresarial.
34. Savedra, Jorge. jorgesaavedra.wordpress.com. jorgesaavedra.wordpress.com/category/patrones-grasp/. [En línea] [Citado el: 20 de Marzo de 2013.]

ANEXOS

Anexo 1: Acta de liberación.



CENTRO PARA LA INFORMATIZACIÓN DE LA
GESTIÓN DE ENTIDADES

Validación y Verificación de Requisitos (V&V)

ACTA DE LIBERACIÓN DEL COMPONENTE
SEGURIDAD Y SALUD EN EL TRABAJO

Acta de Liberación, Versión 1.0 29/ 05/ 2013

Datos del producto**Emitida a favor de:** Seguridad y salud en el trabajo**Fecha de emisión del acta:** 29/05/2013**Responsable:** Ing. Arnolis Salgueiro Arzuaga**Cargo:** Jefe de línea**Clasificado como:** Aplicación Web.

Detalle de los elementos probados y su estado final:

Artefacto	Estado Final
Aplicación	0 No Conformidades
Documentación	0 No Conformidades

Cantidad de iteraciones:

Para la revisión se emplearon un total de 3 iteraciones para lograr el resultado de 0 (cero) No Conformidad.

Elementos revisados o probados y herramientas utilizadas

Elemento	Herramienta
Complejidad	Estándar de interfaz, Lista de Chequeo Interfaz
Ortografía y redacción	Revisión Técnica

Cantidad total de horas empleadas y rango de fechas:

Se emplearon un total de 12 horas efectivas de trabajo con la siguiente distribución: 5 h (17/Mayo/2013), 5h (28/Mayo/2013) y 2 h (29/Mayo/2013).

Estructura del equipo de prueba empleado y turnos de trabajo:

Las pruebas se realizaron en un total de 3 turnos de trabajo, en cada turno se contó con 5 probadores por parte del equipo de pruebas, la actividad estuvo dirigida por un Jefe de Pruebas.

Evaluado por:

Especialista principal Asignado:

Ing. Giselle Almeida González

Otro personal especializado participante:

Nombres y Apellidos	Cargo
Jose Luis Bizet Romero	Estudiante
Roberto Angel Calaña Nalda	Estudiante
Elizabeth Miranda Pérez	Estudiante
Reinier Medinilla González	Estudiante
Yaima Cepero Loyola	Estudiante
Ing. Giselle Almeida González	Administrador de Calidad
Lic. Leandro Roberto Reyes Mejías	Especialista General

 Ing. Giselle Almeida González
 Especialista del Laboratorio de Calidad

 Ing. Arnolis Salgueiro Arzuaga
 Responsable por el Equipo de Desarrollo

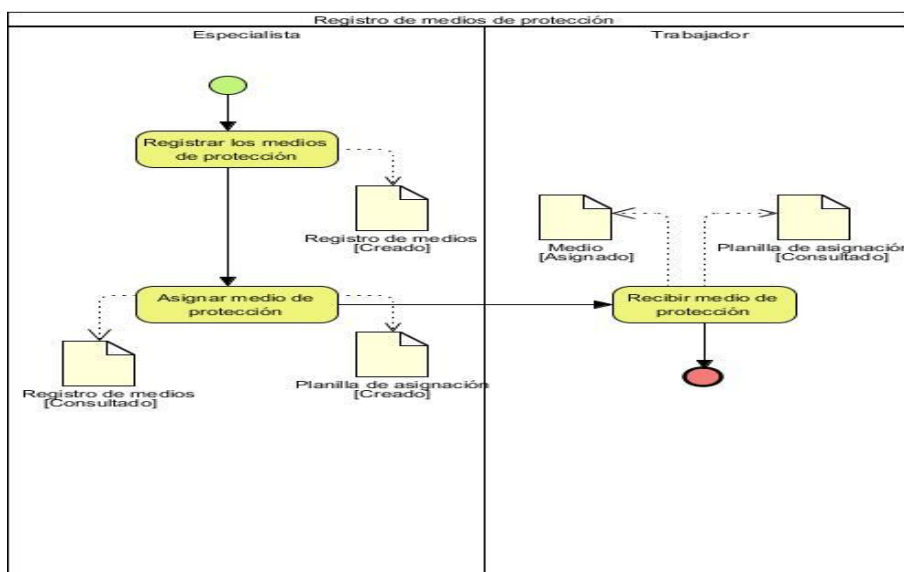
Anexo 2: Descripción del proceso Riesgos Laborales.

Objetivo	Realizar el registro y control de los riesgos laborales en la empresa.
Evento(s) que lo genera(n)	Identificar las situaciones peligrosas.
Pre-condiciones	Se ha definido al menos una situación peligrosa.
Marco legal	<ul style="list-style-type: none"> • Norma Cubana NC 3001-2007 “Sistema de gestión integrada de capital humano” • Norma Cubana NC 3002-2007 “Sistema de gestión integrada de capital humano” • Resolución No. 31/ 2002
Clientes internos	Especialista en Seguridad y Salud.

Clientes externos	NA
Entradas	Listado de situaciones peligrosas.
Flujo de eventos	
Flujo básico Riesgos Laborales	
1	La dirección general identifica las situaciones peligrosas.
2	El especialista en Seguridad y Salud identifica los riesgos asociados.
3	El especialista en Seguridad y Salud identifica las medidas para cada riesgo.
4	El especialista en Seguridad y Salud crea el plan de medidas.
5	La dirección general aprueba el plan de medidas.
6	La dirección general verifica el cumplimiento del plan de medidas.
Pos-condiciones	
1	Se realiza la gestión de los riesgos laborales.
Salidas	
Plan de medidas chequeado.	
Flujos paralelos	
NA	
Pos-condiciones	
NA	
Salidas	
NA	
Flujos alternos	
2.a	
1	
2	
Pos-condiciones	
NA	
Salidas	

NA

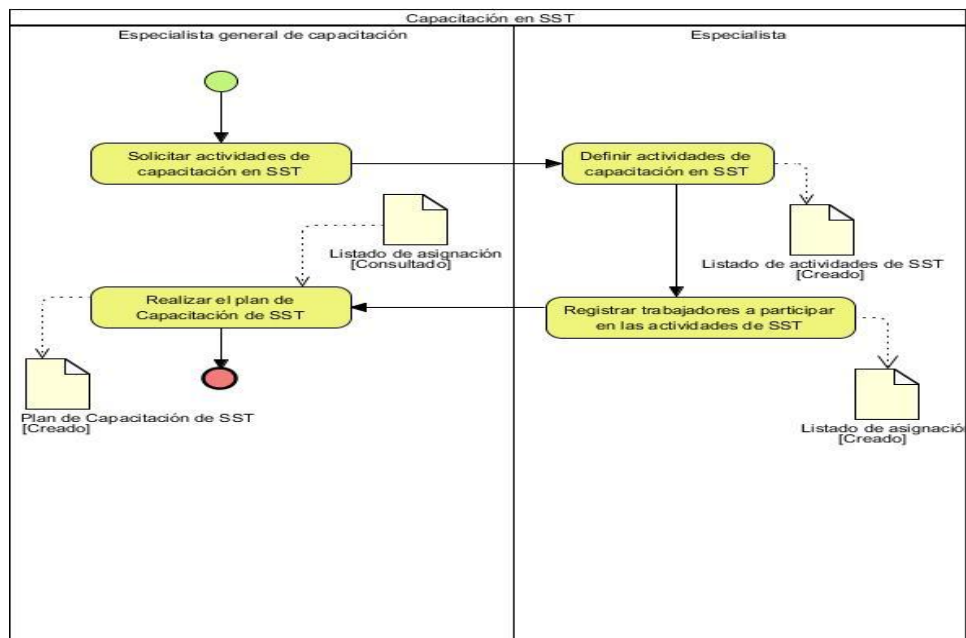
Anexo 3: Diagrama de procesos de Registro de medios de protección.



Objetivo	Realizar el registro y asignación de los medios de protección de la entidad.
Evento(s) que lo genera(n)	NA
Pre-condiciones	<ul style="list-style-type: none"> • Debe existir al menos un trabajador. • Debe existir al menos un puesto de trabajo. • Debe existir al menos un área de trabajo.
Marco legal	<ul style="list-style-type: none"> • Norma Cubana NC 3001-2007 “Sistema de gestión integrada de capital humano” • Norma Cubana NC 3002-2007 “Sistema de gestión integrada de capital humano” • Resolución No. 31/ 2002
Clientes internos	Especialista en Seguridad y Salud. Trabajador

Clientes externos	NA
Entradas	NA
Flujo de eventos	
Flujo básico Riesgos Laborales	
7	El especialista en Seguridad y Salud registra los medios de protección que necesita la entidad.
8	El especialista en Seguridad y Salud asigna los medios de protección al trabajador.
9	El trabajador recibe los medios de protección asignados.
Pos-condiciones	
2	Se realiza el registro de los medios de protección
Salidas	
Planilla de asignación.	
Flujos paralelos	
NA	
Pos-condiciones	
NA	
Salidas	
NA	
Flujos alternos	
2.a	
1	
2	
Pos-condiciones	
NA	
Salidas	
NA	

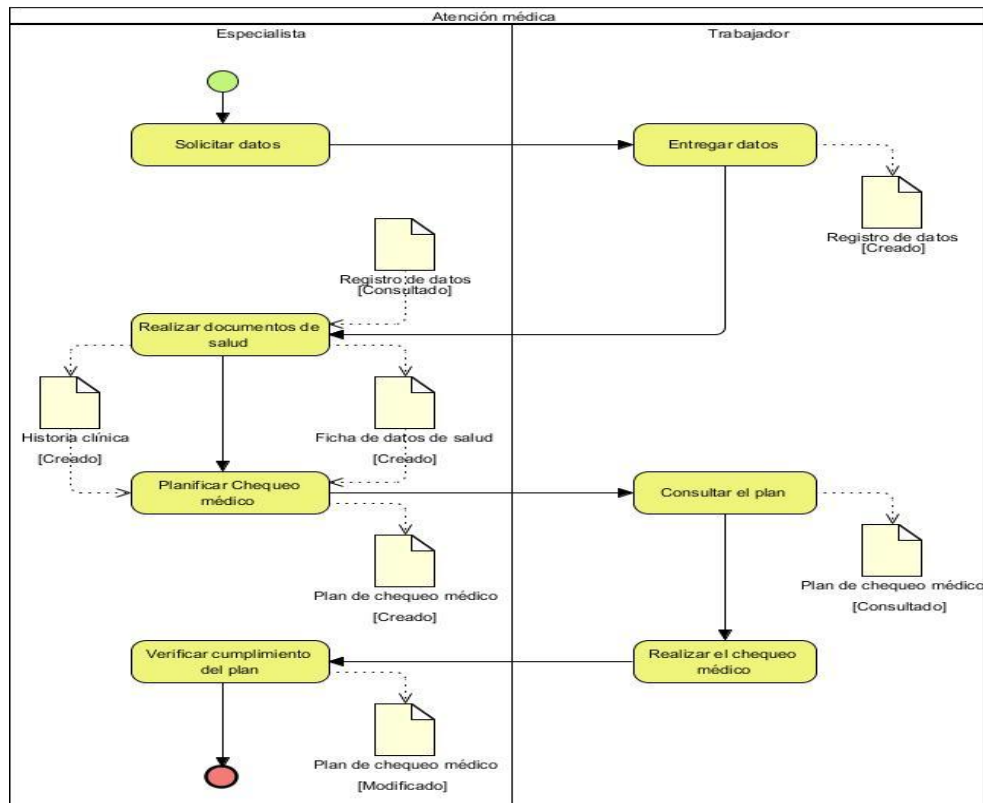
Anexo 4: Diagrama de procesos de Capacitación en SST.



Objetivo	Definir las actividades de capacitación para seguridad y salud.
Evento(s) que lo genera(n)	NA
Pre condiciones	NA
Marco legal	<ul style="list-style-type: none"> • Norma Cubana NC 3001-2007 “Sistema de gestión integrada de capital humano” • Norma Cubana NC 3002-2007 “Sistema de gestión integrada de capital humano” • Resolución No. 31/ 2002
Clientes internos	Especialista en Seguridad y Salud.
Clientes externos	Especialista general de Capacitación.
Entradas	Solicitud de actividades.
Flujo de eventos	
Flujo básico Riesgos Laborales	

10	El especialista en Capacitación solicita las actividades de Seguridad y salud.
11	El especialista en Seguridad y Salud define las actividades de capacitación.
12	El especialista en Seguridad y Salud asigna a los trabajadores a participar en las actividades.
13	Realiza el plan de capacitación.
Pos-condiciones	
3	Se realiza la gestión de las actividades de capacitación en Seguridad y Salud.
Salidas	
Listado de asignación de actividades.	
Flujos paralelos	
NA	
Pos-condiciones	
NA	
Salidas	
NA	
Flujos alternos	
2.a	
1	
2	
Pos-condiciones	
NA	
Salidas	
NA	

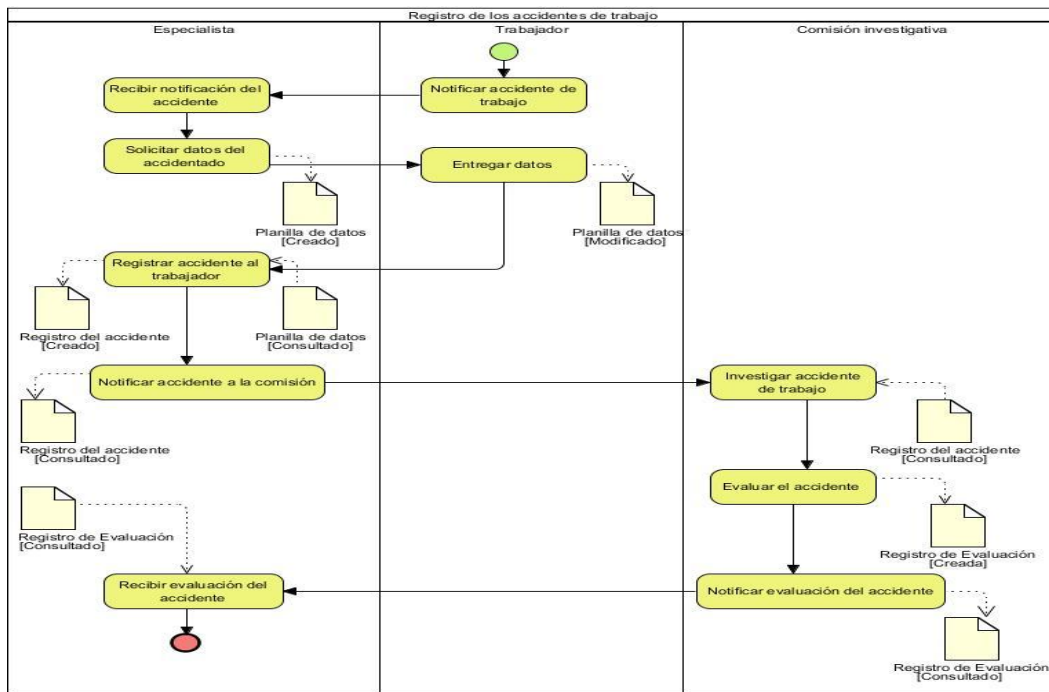
Anexo 4: Diagrama de procesos de Atención médica.



Objetivo	Realizar el registro de los chequeos médicos al trabajador.
Evento(s) que lo genera(n)	NA
Pre-condiciones	NA
Marco legal	<ul style="list-style-type: none"> • Norma Cubana NC 3001-2007 “Sistema de gestión integrada de capital humano” • Norma Cubana NC 3002-2007 “Sistema de gestión integrada de capital humano” • Resolución No. 31/ 2002
Clientes internos	Especialista en Seguridad y Salud. Trabajador.

Clientes externos	NA
Entradas	Datos de Salud del trabajador. Historia Clínica del trabajador.
Flujo de eventos	
Flujo básico Riesgos Laborales	
7	El trabajador Historia clínica y datos de salud al especialista.
8	El especialista en Seguridad y Salud crea el plan de Chequeos médicos
9	El especialista en Seguridad y Salud verifica el cumplimiento del plan de chequeos.
Pos-condiciones	
2	Se realiza la gestión de los chequeos médicos.
Salidas	
	Plan de chequeos médicos.
Flujos paralelos	
NA	
Pos-condiciones	
NA	
Salidas	
NA	
Flujos alternos	
2.a	
1	
2	
Pos-condiciones	
NA	
Salidas	
NA	

Anexo 5: Diagrama de procesos de Registro de los accidentes de trabajo.



Objetivo	Realizar un registro de los accidentes de trabajo.
Evento(s) que lo genera(n)	NA
Pre-condiciones	Se ha notificado al menos un accidente de trabajo.
Marco legal	<ul style="list-style-type: none"> • Norma Cubana NC 3001-2007 “Sistema de gestión integrada de capital humano” • Norma Cubana NC 3002-2007 “Sistema de gestión integrada de capital humano” • Resolución No. 31/ 2002
Clientes internos	Especialista en Seguridad y Salud. Trabajador

	Comisión investigativa.
Clientes externos	NA
Entradas	Planilla de datos del accidentado.
Flujo de eventos	
Flujo básico Riesgos Laborales	
10	El trabajador notifica el accidente.
11	El especialista en Seguridad y Salud registra el accidente.
12	El especialista en Seguridad y Salud notifica el accidente a la comisión.
13	La comisión investiga el accidente.
14	La comisión evalúa el accidente.
15	La comisión notifica evaluación del accidente al especialista.
Pos-condiciones	
3	Se realiza la gestión de los accidentes de trabajo.
Salidas	
	Registro de evaluación del accidente.
Flujos paralelos	
NA	
Pos-condiciones	
NA	
Salidas	
NA	
Flujos alternos	
2.a	
1	
2	
Pos-condiciones	
NA	

Salidas

NA

Anexo 6: Diagramas de clases del diseño.

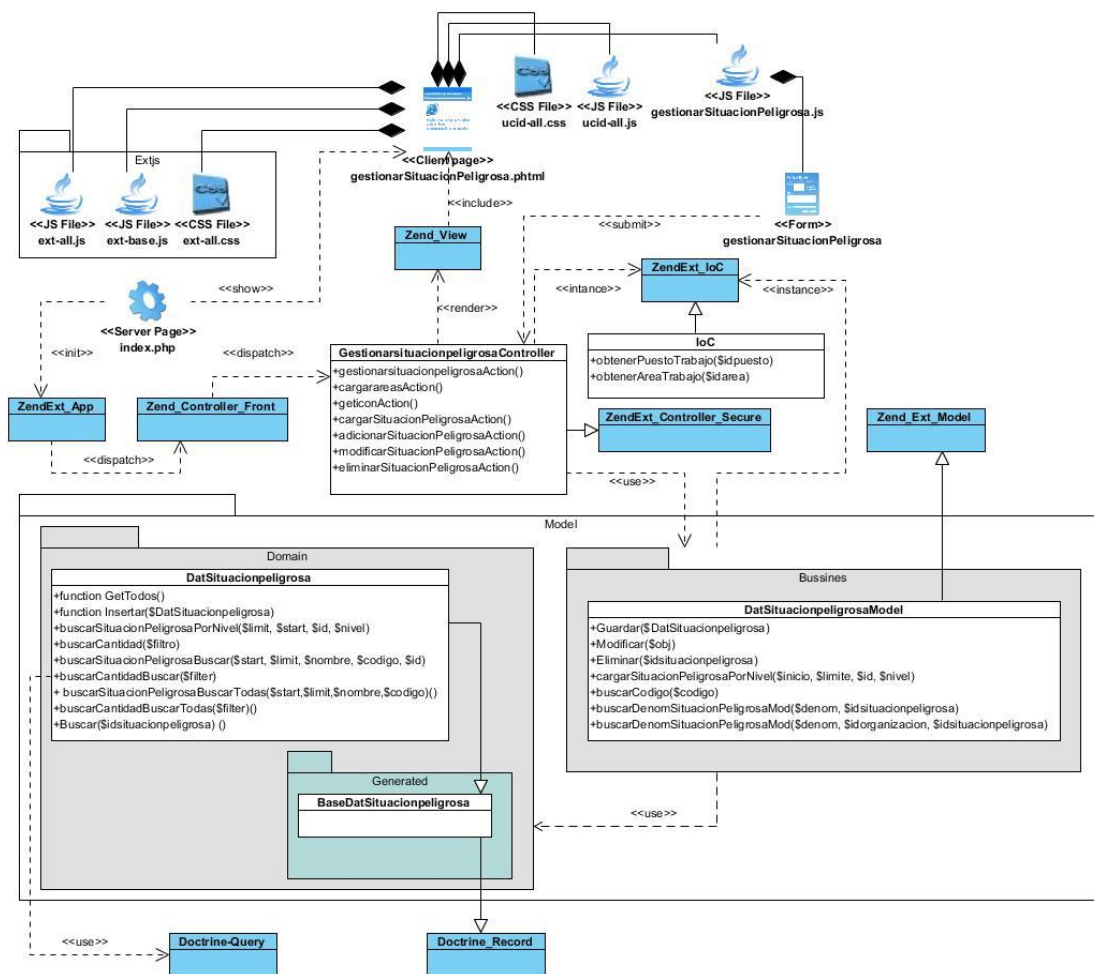


Figura 22: Diagrama de clases del diseño del escenario Gestionar situación peligrosa.

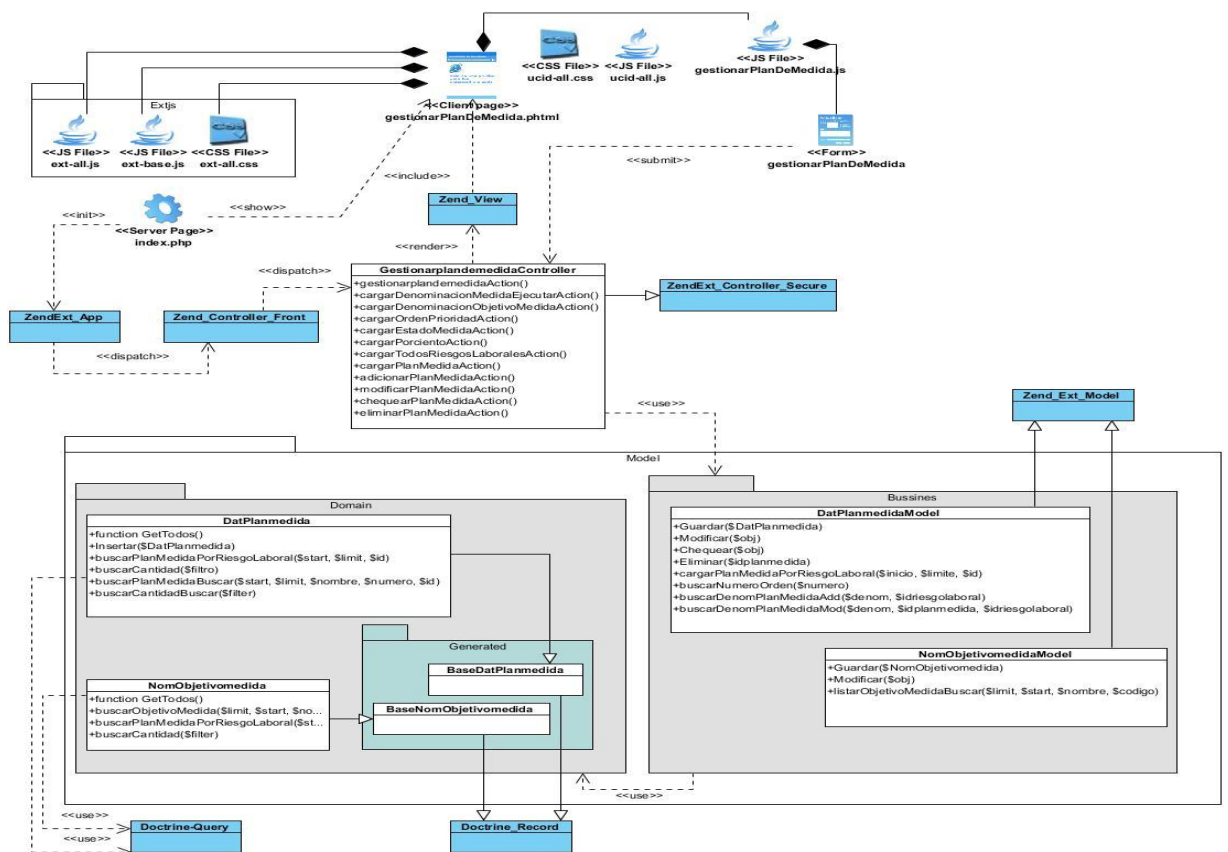


Figura 23: Diagrama de clases del diseño de Gestionar plan de medidas.

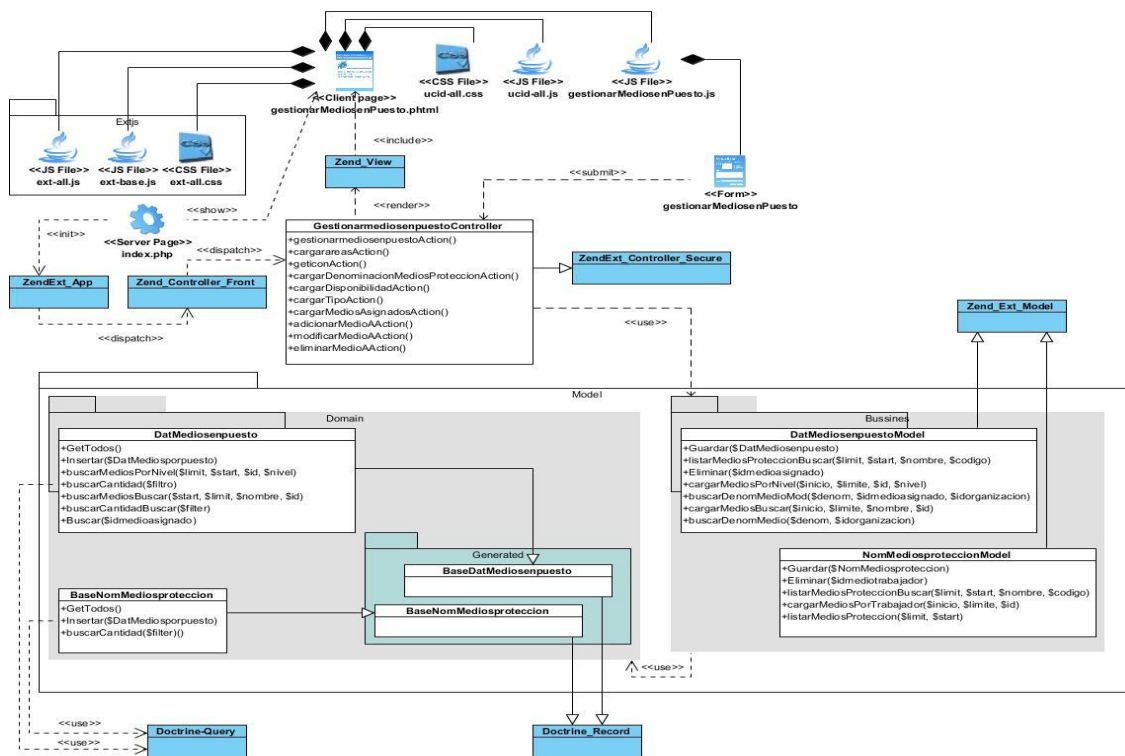


Figura 24: Diagrama de Clases del diseño Registrar medios de protección.

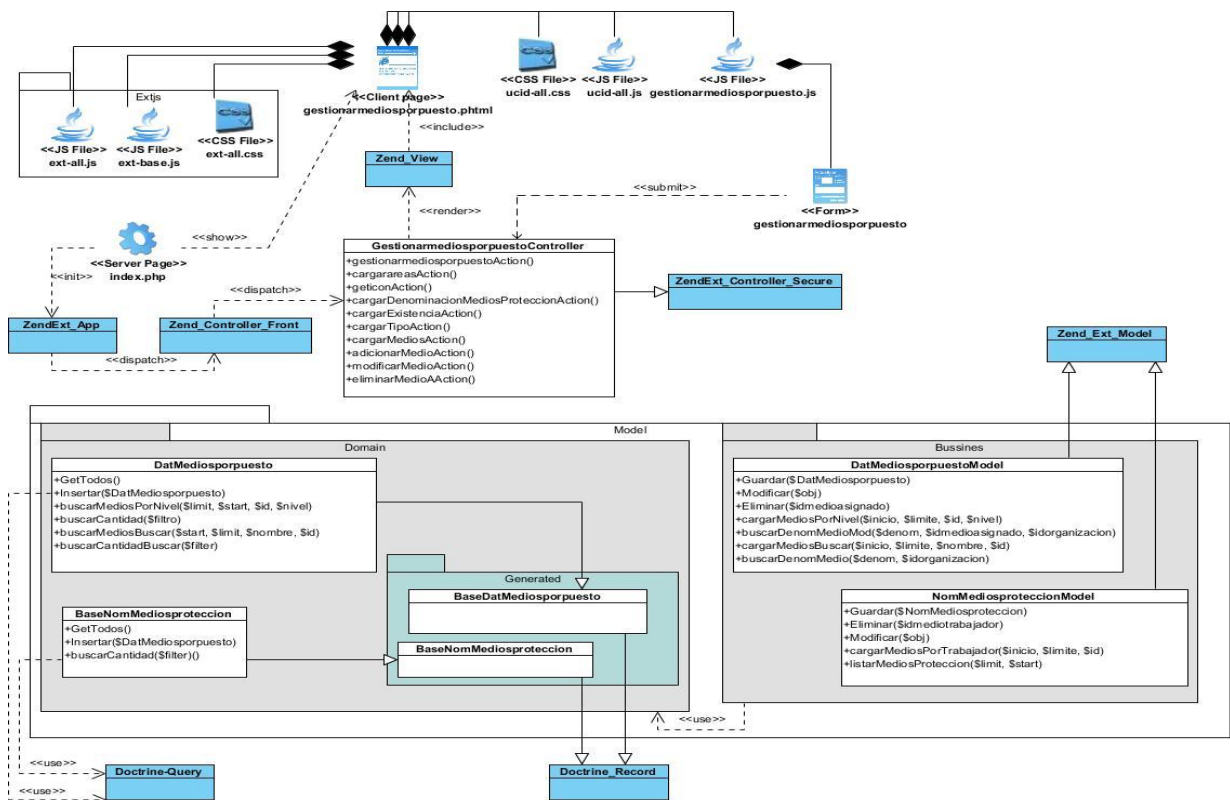


Figura 25: Diagrama de clases del diseño de Asignar medios de protección.

Anexo 7: Diagramas de secuencia.

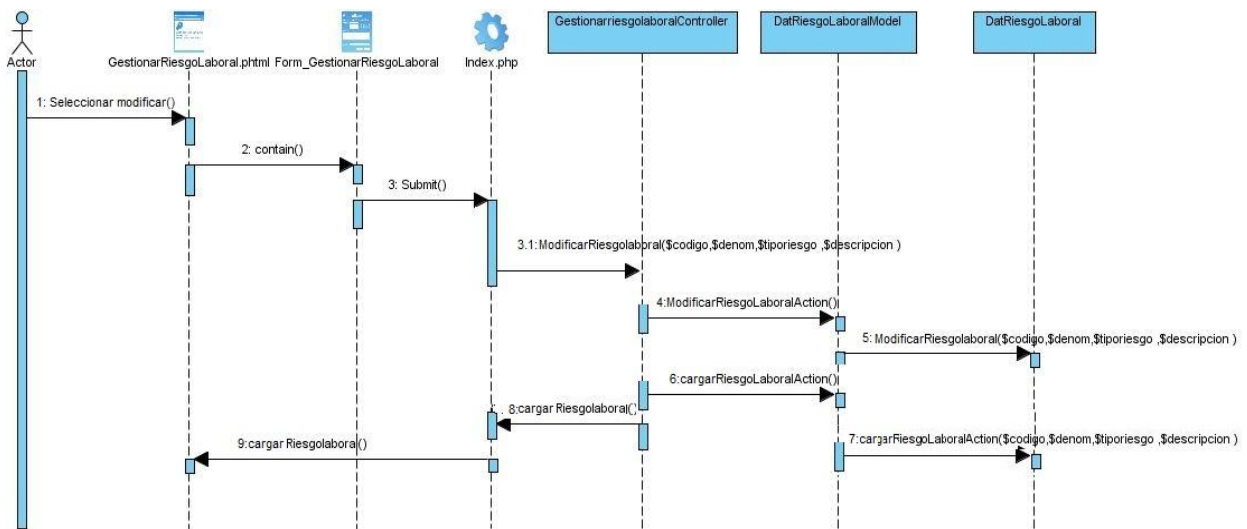


Figura 26: Diagrama de secuencia Modificar riesgo laboral.

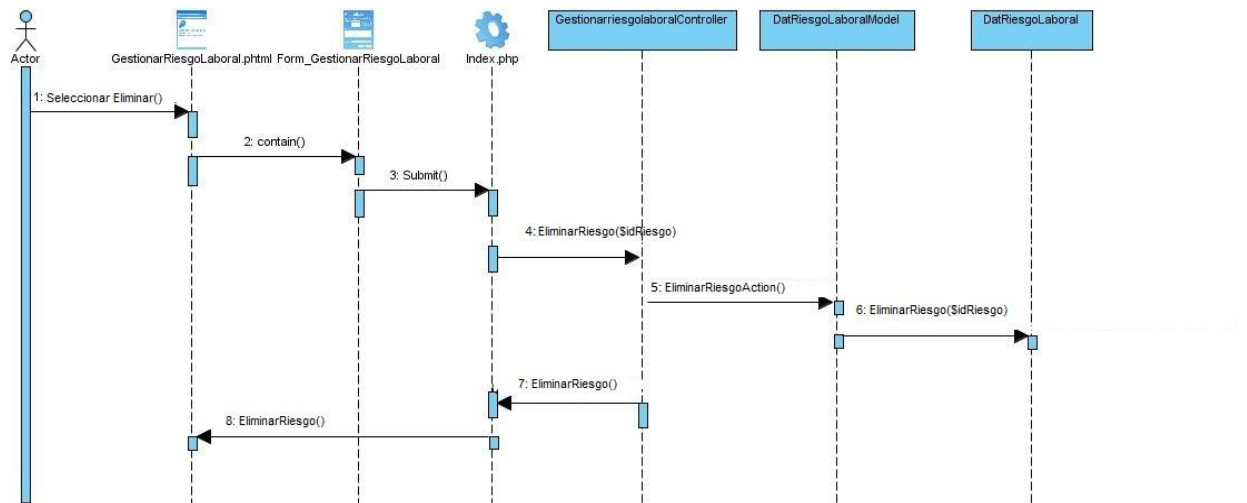


Figura 27: Diagrama de secuencia Eliminar riesgo laboral.

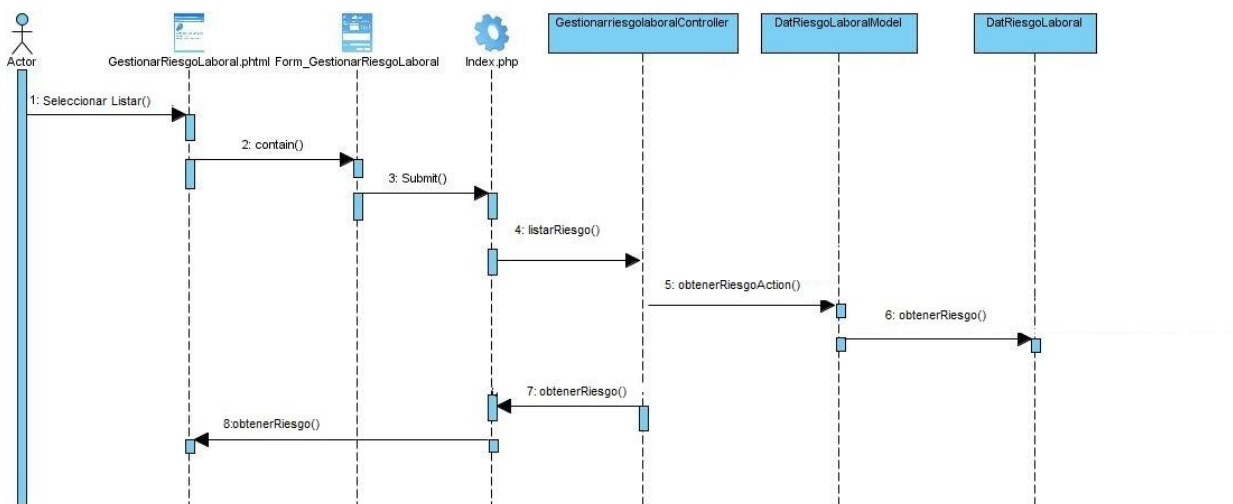


Figura 28: Diagrama de secuencia Listar riesgo laboral.

Anexo 7: Interfaces de implementadas.

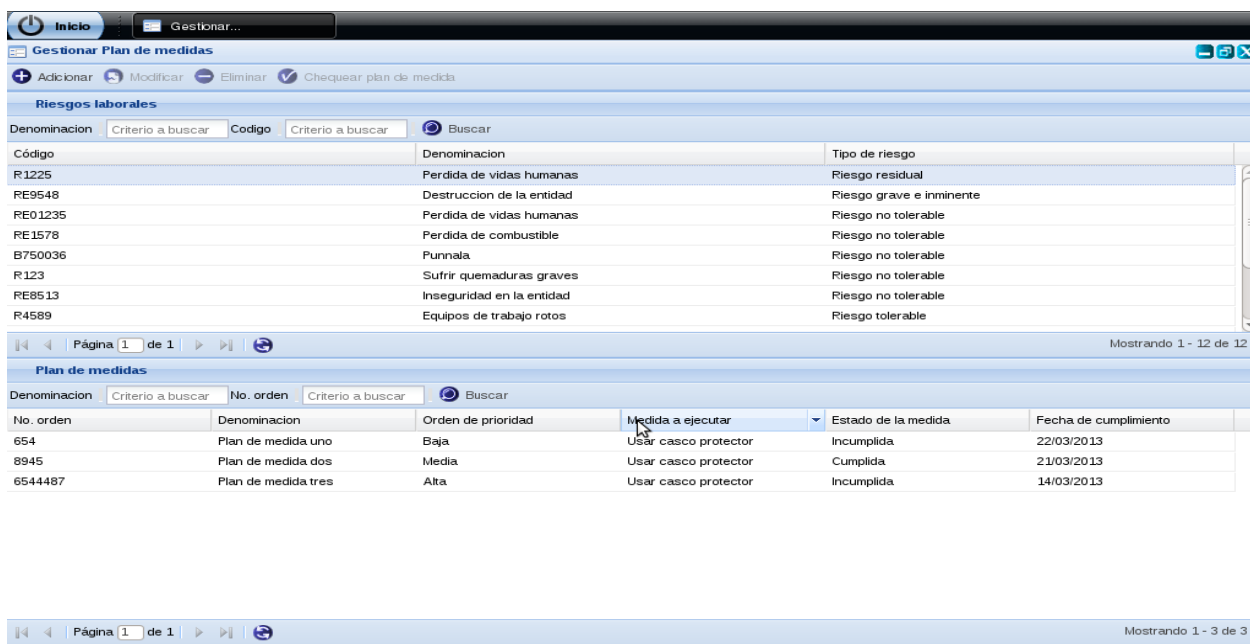


Figura 29: Interfaz Gestionar plan de medidas.

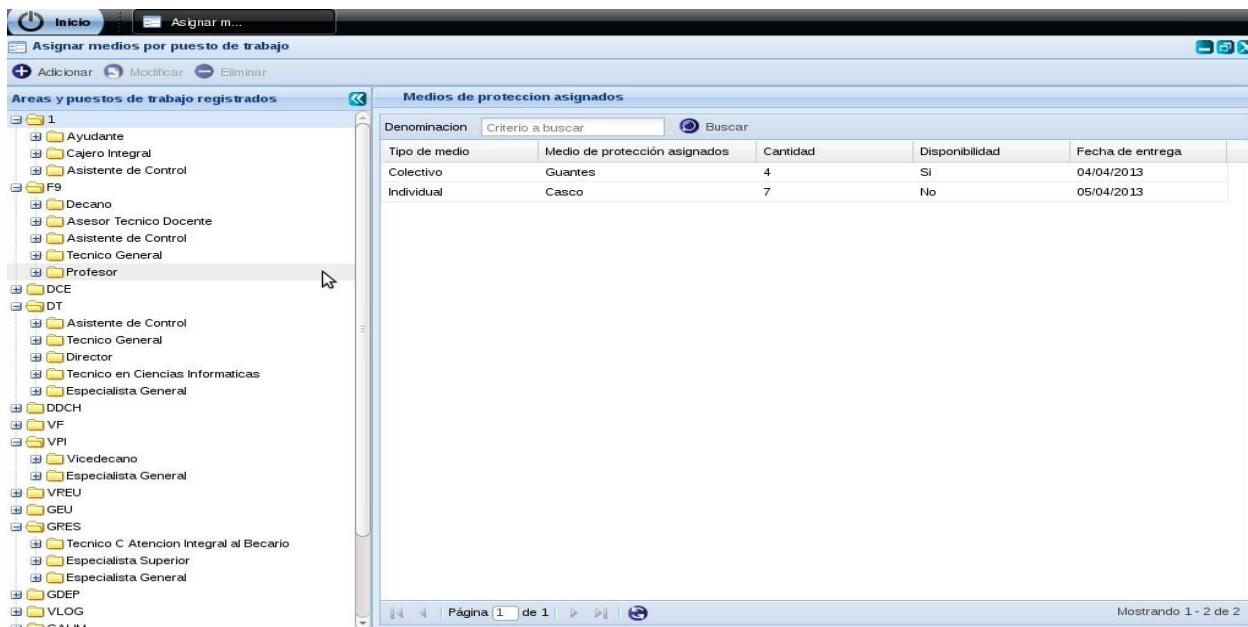


Figura 30: Interfaz Asignar medios por puesto de trabajo.

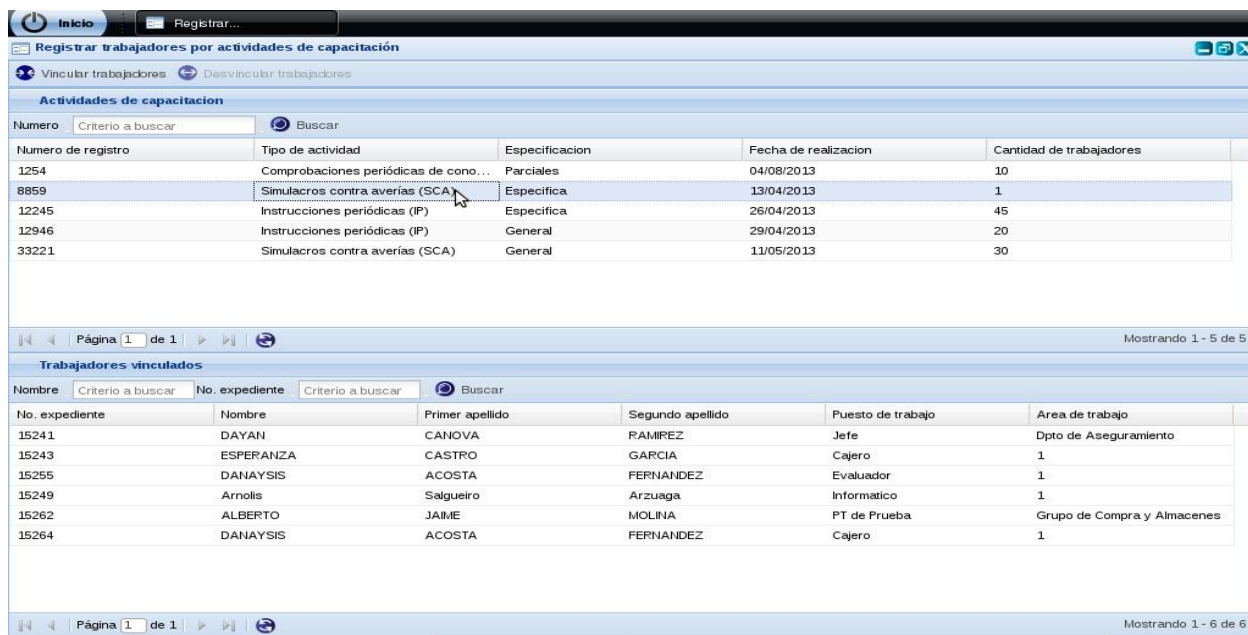


Figura 31: Registrar trabajadores por actividades de capacitación.

Inicio | Gestionar... | Gestionar... | Historia ...

Historia clínica del trabajador

Listado de trabajadores

Nombre Criterio a buscar No. expediente Criterio a buscar Buscar

No. expediente	Nombre	Primer apellido	Segundo apellido	Puesto de trabajo	Area de trabajo
15241	DAYAN	CANOVA	RAMIREZ	Jefe	Dpto de Aseguramiento
15243	ESPERANZA	CASTRO	GARCIA	Cajero	1
15247	JUANAAAAA	Rodriguez	GARCIA	Tia	Grupo de Servicio
15249	Arnolis	Salgueiro	Arzuaga	Informatico	1
15255	DANAYISIS	ACOSTA	FERNANDEZ	Evaluador	1
15262	ALBERTO	JAIME	MOLINA	PT de Prueba	Grupo de Compra y Almacenes
15264	DANAYISIS	ACOSTA	FERNANDEZ	Cajero	1
15271	DANAYISIS	ACOSTA	FERNANDEZ	21	1

Página 1 de 1 | Mostrando 1 - 8 de 8

+ Adicionar... | Modificar... | Eliminar...

Datos de la historia clínica

Numero:	Grado de dispensarizacion:	Talla:	Peso:	Evaluacion nutricional:
5521	dispensado	1.73	71.55	Normal
Grupo sanguineo:	Factor sanguineo:	Fecha de inscripcion:	Genero de vida:	Estado de salud:
A	RH+	05/04/2013	Regular	Bueno

Figura 32: Datos de salud del trabajador.

GLOSARIO

Capital Humano: Cuantificación y valoración de los recursos humanos. Valor de las habilidades, capacidades, experiencias y conocimientos de las personas que integran una organización.

Gestión: Es el conjunto de diligencias que se realizan para desarrollar un proceso o para lograr un producto determinado. Es también la dirección o administración de una empresa o de un negocio.

Entidad: Empresa, unidad presupuestada u otro tipo de organización similar con una gestión económica, financiera, organizativa, técnica, productiva, comercial, laboral y contractual, con autonomía controlada, en cumplimiento de lo establecido por el Gobierno.

Proceso: Según lo establecido en el apartado 3.4.1 de la NC ISO 9000:2005, el proceso es un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados.

Componente: Elemento de un sistema software que ofrece un conjunto de servicios, o funcionalidades, a través de interfaces definidas.

Subsistema: Cada uno de los componentes principales de un sistema que este dividido en componentes. Cada subsistema abarca aspectos del sistema que comparten alguna propiedad común.

Trabajador: Es aquella persona que ocupa un puesto de trabajo en la entidad y desempeña una labor determinada por la cual recibe haberes o salarios.