

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



Facultad 1

Título: Sistema de Gestión para el Control de la Seguridad y Salud en el Trabajo.
Trabajo de diploma para optar por el título de ingeniero en ciencias informáticas

Autor: Harold Pérez Trujillo.

Tutores:

Ing. Enit Martínez González.

Ing. Laritza González Marrero.

Cotutor:

Ing. Daniel Díaz Delgado.

La Habana, mayo de 2021

Declaración de autoría

Se declara que soy el único autor de este trabajo titulado “ Sistema de Gestión para el Control de la Seguridad y Salud en el Trabajo” y se autoriza a la Universidad de las Ciencias Informáticas (UCI) y al Ministerio del Trabajo y Seguridad Social (MTSS) para que hagan el uso que estimen pertinente.

Para que así conste firmo la presente a los __14__ días del mes de __Noviembre__ del año __2021__.



Firma del autor

Harold Pérez Trujillo



Firma de Tutor

Laritza González Marrero

Digitally signed by Enit
Martínez González
Date: 2021.12.15
22:54:51 -05'00'

Firma de Tutor

Enit Martínez González



“Un hombre sabio puede aprender más de una pregunta tonta, que de lo que puede aprender un tonto de una respuesta sabia”.

Bruce Lee

Dedicatoria

Este trabajo va dedicado especialmente a mis padres, Daniel y Jenny que han puesto todo su empeño para que yo pueda ser un profesional, a mi abuelo Roberto que desde el comienzo de mi vida como estudiante siempre me inculcó dedicar empeño y sacrificio a los estudios para lograr todas mis metas, a mi novia por darme fuerzas cada segundo. En general a todos mis familiares, amigos y profesores que me apoyaron en todo momento.

Agradecimientos

Dedico la presente investigación y le doy las gracias a todas las personas que confiaron en mi y me dieron fuerzas para nunca desistir, en especial a mis padres por servir de guía en todo mi proceso de formación tanto de estudiante como en mi vida personal, a mi hermano por apoyarme en todo momento, a mis abuelitos por preocuparse siempre y ayudarme, en especial a mi abuelo Roberto que se que se siente muy orgulloso desde donde esté, de todo el empeño que he puesto para culminar esta investigación, también a mi novia por estar presente y brindarme su ayuda, a toda la familia en general, a mis tutores Enit, Daniel y Laritza, a todos, gracias.

Resumen

En el mundo actual, la Gestión de la Seguridad y Salud en el Trabajo (GSST), ocupa un lugar importante dentro de la gestión de los procesos de Recursos Humanos, pues constituye un pilar fundamental para mantener la fuerza de trabajo satisfecha y altamente motivada. Esto a su vez constituye el eslabón fundamental de toda una cadena de acciones y recursos necesarios para lograr la eficiencia en cualquier organización, partiendo de las insuficiencias en los Sistemas de Gestión de la Seguridad y Salud en el Trabajo (SGSST) en las empresas cubanas, debido a los incidentes y accidentes de trabajo, con repercusiones en la responsabilidad social en las mismas. El aumento de los accidentes laborales ha permitido la necesidad de llevar un mayor control de la seguridad y salud en el trabajo. La presente investigación tiene como objetivo desarrollar una aplicación web que contribuya con el control de la seguridad y salud en el trabajo en Cuba. Por otra parte, aporta a la necesidad de apostar por la soberanía tecnológica como estrategia del país para el desarrollo en todas las esferas. Para facilitar el proceso de desarrollo se utilizó la metodología Programación Extrema (XP), cumpliendo con cada una de sus fases, generando todos los artefactos. El trabajo incluye un estudio del estado del arte, las herramientas utilizadas, las características del sistema, la planificación, el diseño, la codificación y la estrategia de pruebas propuestas por la metodología seleccionada.

Palabras clave: Aplicación web, seguridad y salud en el trabajo, sistema de control, incidentes, accidentes y enfermedades laborales

Índice

Introducción.....	1
Capítulo I. Fundamentación Teórica.....	6
1.1 Conceptos asociados.....	6
1.2 Estudios de Sistemas de Gestión de Seguridad y Salud en el Trabajo.....	7
1.2.1 Internacionales.....	7
1.2.2 Resultados obtenidos por el estudio.....	11
1.3 Análisis de las tecnologías, metodologías, lenguajes y herramientas a utilizar en el desarrollo.....	11
1.3.1 Metodologías de desarrollo.....	11
1.3.2 Lenguajes utilizados.....	13
1.3.3 Entorno de desarrollo.....	19
1.3.4 Sistemas gestores de base datos.....	22
1.3.5 Herramientas CASE.....	24
1.3.6 Conclusiones Parciales.....	25
Capítulo II. Análisis y diseño del sistema.....	26
2.1 Modelo de proceso de negocio.....	26
2.1.1 Descripción del proceso de negocio.....	26
2.2 Roles y responsabilidades.....	27
2.3 Fase de Planeación.....	27
2.3.1 Levantamiento de Requisitos.....	28
2.3.2 Historias de usuario (HU).....	32
2.4 Fase de Diseño.....	34
2.4.1 Tarjetas CRC.....	34
2.4.2 Arquitectura de software.....	35
2.5 Modelo de datos.....	37
2.6 Patrones de diseño.....	38
2.7 Conclusiones Parciales.....	41
Capítulo III. Implementación y prueba del sistema”.....	42
3.1 Tareas de ingeniería.....	42
3.1.1 Diagrama de despliegue.....	45
3.2 Estrategia de pruebas.....	46

3.2.1 Pruebas unitarias.....	46
3.2.2 Pruebas de aceptación.....	48
3.2.3 Resultado de las Pruebas de Aceptación.....	51
3.3 Estándares de codificación.....	52
3.4 Conclusiones Parciales.....	52
Conclusiones Generales.....	53
Recomendaciones.....	54
Referencias bibliográficas.....	55
Bibliografía.....	58
Anexos.....	62
1. Historias de usuario.....	62
2. Tarjetas CRC.....	67
3. Pruebas Unitarias.....	71
4. Resultados de Pruebas de Aceptación.....	72
5. Carta de Aceptación del Cliente.....	73

Índice de Figuras

Figura 1. Diagrama de proceso de negocio.....	26
Figura 2. Patrón arquitectónico MVT [40].....	36
Figura 3. Ejemplo del patrón MVP utilizado en la implementación del sistema.....	37
Figura 4. Modelo de datos.....	38
Figura 5. Código para eliminar un determinado Accidente laboral.....	39
Figura 6. Código para listar los Accidentes laborales.....	39
Figura 7. Código de las URLs principales del proyecto (@login).....	40
Figura 8. Código para habilitar una categoría de usuario.....	40
Figura 9. Diagrama de despliegue.....	46
Figura 10. Test realizado a la clase ProvinciaTestCase.....	47
Figura 11. Test realizado a la clase CategoríaUsuarioTestCase.....	48

Índice de Tablas

Tabla 1. Operacionalización de las variables.....	3
Tabla 2. Resumen de Sistemas Homólogos.....	10
Tabla 3. Casos de usos que responden requisitos funcionales.....	28
Tabla 4. Requisitos no funcionales.....	31
Tabla 5. Ejemplo de Historia de usuario: Autenticar usuario.....	32
Tabla 6. Ejemplo de Tarjeta CRC: Gestionar Usuario.....	35
Tabla 7. Tarea de ingeniería 1: Autenticación de Usuario.....	42
Tabla 8. Tarea de ingeniería 2: Registrar Usuario.....	43
Tabla 9. Tarea de ingeniería 3: Modificar Categoría de Usuario.....	43
Tabla 10. Tarea de ingeniería 4: Eliminar Accidente Laboral.....	43
Tabla 11. Tarea de ingeniería 5: Detalles de Persona.....	44
Tabla 12. Tarea de ingeniería 6: Buscar Peligro.....	44
Tabla 13. Tarea de ingeniería 7: Eliminar Riesgos Seleccionados.....	45
Tabla 14. Prueba de Aceptación: Autenticar usuario.....	49
Tabla 15. Prueba de Aceptación: Gestionar usuario.....	49
Tabla 16. Prueba de Aceptación: Gestionar accidente laboral.....	49
Tabla 17. Prueba de Aceptación: Gestionar persona.....	50
Tabla 18. Prueba de Aceptación: Gestionar peligro.....	50
Tabla 19. Prueba de Aceptación: Gestionar riesgo.....	51

Introducción

La constitución de la Organización Internacional del Trabajo (OIT) establece el principio de que los trabajadores deben estar protegidos contra las enfermedades en general, o las enfermedades profesionales y los accidentes resultantes de su trabajo. No obstante, para millones de trabajadores ello dista mucho de ser una realidad. De conformidad con las estimaciones globales más recientes de la OIT, cada año se producen 2,78 millones de muertes relacionadas con el trabajo, de las cuales 2,4 millones están relacionadas con enfermedades profesionales. Además del inmenso sufrimiento que esto causa a los trabajadores y sus familias. Los costes económicos que ello conlleva son enormes para las empresas, los países y el mundo en general. Las pérdidas relacionadas con las indemnizaciones, las jornadas laborales perdidas, las interrupciones de la producción, la formación y la readaptación profesional, y los costes de la atención sanitaria representan alrededor del 3,94 por ciento del Producto Interno Bruto (PIB) (mundial [1]). Para los empleadores, esto se traduce en costosas jubilaciones anticipadas, pérdida de personal cualificado, ausentismo y altas primas de seguro. Sin embargo, estas tragedias podrían evitarse con la adopción de métodos racionales de prevención, notificación e inspección.

Las normas de la OIT sobre la seguridad y salud en el trabajo proporcionan a los gobiernos, empleadores y trabajadores los instrumentos necesarios para desarrollar tales métodos y garantizar la máxima seguridad en el trabajo. La OIT ha adoptado más de 40 normas que tratan específicamente de la seguridad y la salud en el trabajo, así como más de 40 repertorios de recomendaciones prácticas. Cerca de la mitad de los instrumentos de la OIT tratan directa o indirectamente de cuestiones relativas a la seguridad y a la salud del trabajo.

Empresas de todo el mundo están cada vez más involucradas en el alcance y la demostración de un sólido desempeño de la Seguridad y Salud en el Trabajo (SST) mediante el control de sus riesgos, de acuerdo con su política y objetivos de SST. Esto se lleva a cabo en un contexto legislativo cada vez más exigente, con el desarrollo de políticas económicas y otras medidas para fomentar las buenas prácticas de SST, y de una mayor preocupación de las partes interesadas en este tema. La incorporación de Tecnologías de la Información y las Comunicaciones (TIC) en la actividad de la SST, se han convertido en una herramienta que aporta una nueva perspectiva sobre el tratamiento de la gestión del riesgo y la oportunidad que se tiene para poder incidir sobre él a partir del uso de las mismas.

En América Latina hay retos importantes relacionados con la salud y seguridad en el trabajo. Las cifras indican que se registran 11,1 accidentes mortales por cada 100.000 trabajadores en la industria, 10,7 en la agricultura, y 6,9 en el sector de los servicios [2]. Algunos de los sectores más importantes para las economías de la región, como la minería, construcción, agricultura y pesca, se encuentran también entre aquellos en los cuales se produce la mayor incidencia de accidentes. Para la OIT es importante que los países de América Latina y el Caribe cuenten con un marco normativo adecuado, que tengan políticas nacionales y programas de salud y seguridad en el trabajo, y que promuevan la acción coordinada de las diferentes entidades que tienen que ver con estos temas. De la misma manera, se ha planteado que la existencia de un sistema de inspección eficaz para velar por el cumplimiento de la norma es clave.

En Cuba la seguridad y la salud en el trabajo han sido temas de interés en las diferentes etapas del desarrollo histórico de la sociedad, por lo que la formalización de sus métodos y fines, así como su cuerpo teórico, son el resultado de la

producción investigativa de profesionales de diferentes especialidades. Sin embargo, todavía existen limitaciones que deben ser superadas con la misma rapidez con que se han desarrollado en los siglos XX y XXI, caracterizados por las nuevas necesidades derivadas del empleo de las TIC.

En ocasiones, los sistemas de gestión de la seguridad y salud en el trabajo implementados en las empresas del país, no cumplen con el objetivo de disminuir o evitar la ocurrencia de incidentes, accidentes de trabajo y enfermedades profesionales. Aspecto que demuestra las insuficiencias en la implementación de este sistema en las empresas cubanas, al igual que la inadecuada aplicación de la Responsabilidad Social Empresarial (RSE). Estas deficiencias y otras que persisten en la economía, constituyeron los fundamentos para la elaboración de los lineamientos de la política económica y social, aprobados en el VI Congreso del Partido Comunista Cubano en 2011.

Por otra parte, el bloqueo económico, comercial y financiero impuesto por los Estados Unidos contra Cuba desde hace más de 60 años, afecta severamente la seguridad y la salud en el trabajo, debido a que en mucho de los casos no se pueden adquirir materiales y herramientas de trabajo primordiales para la protección y seguridad de los trabajadores en las diferentes entidades.

La seguridad y la salud en el trabajo han sido abordadas en Cuba desde diferentes aristas que constituyen aspectos más trascendentales a la exposición de riesgos laborales. Por tal motivo, el estudio de los accidentes de trabajo, el ambiente laboral y la morbilidad laboral temporal han sido temas investigados en las tesis defendidas en la maestría en salud ocupacional, durante el periodo comprendido entre 1984-2006.[3]

El estado cubano tiene la responsabilidad de garantizar mediante la asignación y planificación, los recursos materiales y financieros necesarios para crear las condiciones necesarias de la SST. El Sistema de Gestión de la Seguridad y Salud en el Trabajo (SGSST) va encaminado a garantizar las condiciones de seguridad y salud de los trabajadores. Lo que le permite actuar como factor de motivación y eficiencia de los trabajadores sobre la base de integrar sus principios y tareas a los sistemas de gestión y en general a las distintas actividades y funciones de la empresa. Factor que incide en la responsabilidad social. Las malas condiciones de trabajo provocan lesiones a los trabajadores, pérdidas de bienes materiales y deterioro del clima laboral, además de la poca productividad; por tanto, un eficiente SGSST es aquel donde exista una correcta armonía entre sus elementos configuradores.

El Ministerio del Trabajo y Seguridad Social (MTSS), ubicado en la provincia La Habana, se extiende por todas las provincias del país, y en cada una de ellas hay una Dirección Provincial del Trabajo y la Seguridad Social (DPTSS). De la misma forma, en todos sus municipios se encuentran una Dirección Municipal del Trabajo y la Seguridad Social (DMTSS). Actualmente el MTSS realiza esta gestión de forma no automatizada debido a que no cuenta con un sistema de este tipo, lo que ha conllevado a desarrollar un Sistema de Gestión de la información relacionada con los accidentes y peligros para el Control de la Seguridad y Salud en el Trabajo del país.

Teniendo en cuenta la situación descrita anteriormente se arriba al siguiente **problema de investigación:** ¿Cómo contribuir a la gestión de la información relacionada con los accidentes y peligros en el marco de la seguridad y la salud en el trabajo en Cuba?

Siendo el **objetivo general**: Desarrollar un sistema para informatizar el proceso de gestión de la información relacionada con los accidentes y peligros para el control de la seguridad y salud en el trabajo en el MTSS, que se desglosa en los siguientes

Objetivos específicos:

1. Elaborar el marco teórico de la investigación sobre soluciones informáticas asociadas a la gestión de los accidentes y peligros en Sistemas de control de seguridad y salud en el trabajo.
2. Diseñar un sistema que permita la gestión de los accidentes y peligros para el Sistema de control de seguridad y salud en el trabajo.
3. Implementar un sistema que permita la gestión de la información relacionada con los accidentes y peligros para este Sistema.
4. Evaluar el sistema desarrollado, aplicándole las pruebas necesarias .

Como objeto de estudio se propone el proceso de Gestión de la Seguridad y Salud en el trabajo, se define como **campo de acción** el Sistema de Gestión para el Control de la Seguridad y Salud en el Trabajo.

Tras definir los objetivos y analizar el problema, se obtiene la siguiente **hipótesis**: el desarrollo de un Sistema de Gestión para el Control de la Seguridad y Salud en el Trabajo, mejora la Gestión para el Control de la Seguridad y Salud en el Trabajo en el MTSS.

Identificándose como variable independiente: Sistema de Gestión para el Control y como variable dependiente: la Gestión para el Control de la Seguridad y Salud en el Trabajo, como se muestra en la Tabla 1.

Tabla 1. Operacionalización de las variables

Variable conceptual	Dimensión	Indicadores	Sub-indicadores	Unidad de medida
Módulo de Gestión de Accidentes y Peligros para el Sistema Informático	Ministerio del Trabajo y la Seguridad Social	Funcionalidad	Funcionalidad	(%)
		Usabilidad	Instructibilidad	(%)
			Operatividad	(%)
Mejora la	Usuarios que	Fiabilidad e	Alta	95-100 (%)

Gestión de Accidentes y Peligros para el control de la seguridad y salud en el trabajo	utilizan el módulo	integridad de los datos registrados	Media	85-95 (%)
			Baja	0-85 (%)
	Tiempo de búsqueda de una incidencia de trabajo	Alta	0-5 (min)	
		Media	5-15 (min)	
		Baja	Más de 15 (min)	

Se declaran como **tareas de la investigación:**

1. Definición del marco teórico de los accidentes y peligros en la seguridad y salud en el trabajo.
2. Análisis de los sistemas para el control de la seguridad y salud en el trabajo, enfocados en la gestión de accidentes y peligros.
3. Selección de la metodología, tecnologías, lenguajes y herramientas para el desarrollo del sistema.
4. Análisis y diseño de la propuesta de solución.
5. Implementación de la propuesta de solución.
6. Diseño de pruebas de software y de validación de la solución implementada.

Los métodos de investigación empleados son:

Métodos teóricos:

El método Histórico-Lógico permitió el estudio y la comprensión del estado y las tendencias actuales del proceso de control de la seguridad y la salud en el trabajo.

El método Analítico-Sintético se utilizó en la búsqueda y análisis de las herramientas para el control de la seguridad y salud en el trabajo.

La modelación permitió una mayor comprensión de los elementos analizados y diseñados para la implementación de diagramas y modelos más sencillos.

Métodos empíricos:

La Entrevista se empleó en encuentros con los jefes del departamento de la seguridad y la salud en el trabajo del MTSS, la cual brindó una mayor facilidad del proceso de la información, así como obtener el listado de requisitos y sus descripciones.

La presente investigación está estructurada en 3 capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos que aportarán una mayor amplitud de la información que se genera en la investigación.

Capítulo 1: “Fundamentación teórica”. En este capítulo se definen los conceptos que ayudarán a comprender la investigación. Se realiza un estudio y una valoración sobre algunos sistemas para el control de la seguridad y la salud en el trabajo, se explican las herramientas, las metodologías y el lenguaje que será utilizado para construir la solución.

Capítulo 2: “Análisis y diseño del sistema”. Analiza y presenta la solución que se propone, se dan a conocer elementos de cómo está concebido el negocio y se realiza la selección de los requerimientos del sistema que se implementan. Se modelan y describen los diagramas que representan las funcionalidades del sistema, aplicando los patrones de arquitectura y diseños seleccionados.

Capítulo 3: “Implementación y prueba del sistema”. Por último, se describe la herramienta propuesta mediante el apoyo de los diagramas de componentes y despliegue desde el enfoque de la programación. Se muestran los casos de pruebas para comprobar que el software cumple con los requerimientos establecidos. Al igual que la implementación y la validación realizada posteriormente al producto que se obtiene como solución.

Capítulo I. Fundamentación Teórica.

En el presente capítulo se realiza la explicación de diferentes conceptos relacionados al tema de investigación como sistema de control, seguridad y salud en el trabajo e incidentes, accidentes y enfermedades laborales. Se explica la selección de las herramientas, lenguajes y metodología, entorno de desarrollo (IDEs) y artefactos a generar. Además, se estudian los sistemas ya existentes que gestionan la seguridad y salud en el trabajo.

1.1 Conceptos asociados

Sistema de gestión empresarial

Es la parte de la gestión general de la organización que determina y aplica la política integrada de gestión, y que surge de la integración de las gestiones de la calidad, el medio ambiente y la seguridad y salud del trabajo.[4]

Sistema de control

Un sistema de control interno es parte importante de una entidad bien organizada, pues garantiza salvaguardar los bienes y hacer confiable los registros y resultados de la contabilidad; acciones indispensables para la buena marcha de cualquier organización, grande o pequeña [5]. Conjunto de elementos que, ordenadamente relacionados entre sí conllevan a determinado objetivo.

Seguridad y salud en el trabajo

La seguridad y salud en el trabajo (SST) tienen como objetivos, garantizar condiciones seguras e higiénicas, prevenir los accidentes, enfermedades profesionales y otros daños a la salud de los trabajadores y al medio ambiente laboral [6]. Es un área de muchas disciplinas, relacionadas con la seguridad, salud y la calidad de vida de las personas en la ocupación y a su vez protege a toda persona que pueda verse afectada por el ambiente laboral.

Incidentes, accidentes y enfermedades laborales

Se denomina incidente al suceso acaecido en el trabajo o en relación con este, con posibilidad de convertirse en accidente de trabajo u otros daños, en el que la

persona afectada no sufre lesiones corporales o estas no afectan su capacidad para el trabajo.[7]

El accidente de trabajo es un hecho repentino relacionado casualmente con este, que produce al trabajador una lesión corporal que afecta su capacidad para laboral por una o varias jornadas de trabajo, o la muerte.[8]

La enfermedad laboral es la alteración de la salud, patológicamente definida, generada por razón de la actividad laboral en trabajadores que en forma habitual se exponen a factores que producen enfermedades y que están presentes en el medio laboral o en determinados cargos y que es reconocida en la legislación vigente.[9]

1.2 Estudios de Sistemas de Gestión de Seguridad y Salud en el Trabajo

Existen sistemas informáticos que realizan el proceso de gestión de SST en las diferentes empresas del mundo, ejemplo de ellos son GESTIÓN SALUD Y SEGURIDAD EN EL TRABAJO (apk), NOVASOFT-GTH y SARA - Sistema de Seguridad y Salud en en Trabajo. En el ámbito nacional, el MTSS no cuenta con ningún programa informático que lleve el control de este proceso y se desconoce de este tipo de sistemas en el país.

1.2.1 Internacionales

GESTIÓN SALUD Y SEGURIDAD EN EL TRABAJO (apk):

Esta aplicación android gratuita, es una biblioteca dinámica impulsada por los mejores sitios web educativos de franceses especializados en GESTIÓN SALUD Y SEGURIDAD EN EL TRABAJO. Creada por Jean Charles Yuan y publicada en 2019. Contiene muchas opciones como videos, cursos de Prevención de riesgos laborales, manejo de SG-SST, charlas de inducción de seguridad, seguridad privada e industrial, además de Sistemas de Gestión [10]. Esta apk nos brinda información valiosa acerca del tema que se lleva en la investigación, pero no es factible su uso porque no cubre la gestion de la seguridad y salud, debido a que no hace uso de la gestión de peligros y accidentes, solo es una biblioteca de información de la seguridad y salud en el trabajo, además de ser una aplicación android y se requiere de una aplicación web para facilitar la interacción con el usuario.

NOVASOFT-GTH:

Software que permite administrar el ciclo Planear, Hacer, Verificar, Actuar (PHVA) del Sistema de Gestión de Seguridad y Salud en el Trabajo - SG-SST, dando cumplimiento a la normativa colombiana y brindando la información necesaria para fines internos y uso de terceros, permitiéndole desarrollar planes que reducen en el bienestar de los trabajadores y la reducción de riesgos. Cuenta con la posibilidad de generar la planificación inicial, crear los planes anuales, gestionar la matriz de peligros y riesgos, definir la matriz de requisitos legales, ejecutar y hacer seguimiento de actividades tales como [11]:

- Capacitaciones
- Gestión de accidentes e incidentes.
- Gestionar comités (COPASST, BRIGADA, COMITÉ DE CONVIVENCIA).
- Hacer seguimiento al estado de salud de los trabajadores.
- Mantener actualizado el perfil sociodemográfico.
- Consolidar la administración de documentos.
- Gestionar planes de acción.
- Generar mediciones de resultados a través de indicadores de gestión y reportes.

A pesar de estar desarrollado en una tecnología web, no cumple totalmente con el funcionamiento requerido por el sistema, sería de gran utilidad la Gestión de accidentes e incidentes y la medición de resultados a través de indicadores de gestión y reportes, pero la gestión de entidades no tendría un uso factible debido a que Colombia no cuenta con las mismas entidades que nuestro país, además de ser un software de corte privativo lo cual no se puede acceder sin pago de licencia, por otra parte nuestro país esta apostando por la soberanía tecnológica y gustaría de tener un software de nuestra propiedad.

SARA - Sistema de Seguridad y Salud en el Trabajo:

Se adapta y cumple con los requisitos que exige la normativa del sistema de Seguridad social en Colombia.

Está concebido como una herramienta que permite planear, organizar, controlar y realizar gestión sobre los aspectos de la salud ocupacional manejados por las organizaciones de cualquier sector económico, tales como [12]:

- Riesgos profesionales.

- Higiene y seguridad industrial.
- Evaluación y seguimiento médico de los empleados y accidentalidad laboral entre otros.

Agiliza el proceso de administración, control y gestión de salud ocupacional en la Organización. Integra los aspectos más relevantes de la salud ocupacional como Riesgos Profesionales, Medicina Laboral, Higiene y Seguridad Industrial, Accidentalidad, permite promover relaciones y generar información gerencial de cada uno de ellos. Por ser un software de plataforma web, permite que sea de fácil instalación y actualización, aprovechando las ventajas de esta tecnología, como la rapidez y oportunidad de consulta. Adaptable y parametrizable a cualquier tipo de negocio y a cambios gubernamentales o políticas empresariales. Permite la entrada y salida de datos, mediante archivos o interfaces, desde y hacia otros módulos o sistemas de información. Accede al manejo de múltiples clases de informes, reportes y consultas de cada una de las actividades de salud ocupacional; así como la generación de indicadores de gestión de cada uno de los procesos o actividades de salud ocupacional ejecutadas.[12]

Software desarrollado en tecnología web, que cumple parcialmente los funcionamientos requeridos por el sistema, donde la gestión de riesgos profesionales podría ser de utilidad, si bien es adaptable y parametrizable a cualquier tipo de negocio y cambios gubernamentales o políticas empresariales, es un software de pago por licencia lo que lo dificulta y Cuba apuesta por la soberanía tecnológica y la necesidad de tener un software desarrollado por nuestro país para seguir con la continuidad y el desarrollo tecnológico alcanzado en los últimos tiempos.

El análisis a los SG-SST permitió elaborar un resumen en forma de tabla, como se muestra en la Tabla 2, teniendo en cuenta los siguientes parámetros:

- **Adquisición:** Referente al costo de adquisición del producto. Puede tomar valores como: Pago de licencia (cuantía o cuota por la adquisición del producto). Re-ingeniería (se cuenta con toda la documentación necesaria para desarrollarla), Libre (es de uso público y gratuito).
- **Funcionalidad:** Referente a la cobertura de los requisitos funcionales que da la aplicación. Puede tomar valores como: Total cuando cubre el total de las funcionalidades requeridas, Parcial cuando cubre parcialmente las

funcionalidades requeridas, Ninguna cuando no cubre las funcionalidades requeridas.

- **Web:** Referente a si la herramienta es un aplicativo web o no.

Ventajas de una web:

1. Presencia en línea 24/7.
2. Intercambio de información.
3. Credibilidad.
4. Reduce costos.
5. Expansión del mercado.
6. Análisis del consumidor.
7. Publicidad.
8. Competitividad.
9. Servicio al cliente en línea.
10. Oportunidad de crecimiento.

Tabla 2. Resumen de Sistemas Homólogos

Herramienta	Adquisición	Funcionalidad	Web
GESTIÓN SALUD Y SEGURIDAD EN EL TABAJO (apk)	Libre	Ninguna	No
NOVASOFT-GTH	Pago de licencia	Parcial	Si
SARA - Sistema de Seguridad y Salud en en Trabajo	Pago de licencia	Parcial	Si

1.2.2 Resultados obtenidos por el estudio.

El autor arriba a la conclusión de que los sistemas señalados no cumplen totalmente con el objetivo de la presente investigación. Ninguno de ellos reúne los requisitos necesarios para ser utilizados directamente como propuesta de solución, por lo que debe desarrollarse una nueva. A pesar de ello cada uno contribuyó a una mejor comprensión de las principales funcionalidades que debe cumplir un sistema informático de este tipo, como son el caso de los gestionar la información de la Seguridad y Salud en el Trabajo, hacer reportes de ellas, además implementar filtros de búsquedas para facilitar el trabajo con el sistema. Tanto el sistema NOVASOFT-GTH como SARA no cumplen totalmente porque son sistemas privativos, también tendríamos que llenar los datos directamente a internet y eso es información confidencial de nuestro país, pero tienen características muy importantes y las asumo para la solución. Por otra parte nuestro país apuesta por la soberanía tecnológica por lo que surge la necesidad de contar con un software propio.

1.3 Análisis de las tecnologías, metodologías, lenguajes y herramientas a utilizar en el desarrollo.

1.3.1 Metodologías de desarrollo

En la actualidad, la rapidez y el dinamismo en la industria del software, han hecho replantear los cimientos sobre los que se sustenta el desarrollo de software tradicional. Estudios recientes y el mismo mercado actual, están marcando la tendencia en la ingeniería del software teniendo como características principales atender a las necesidades de rapidez y flexibilidad. Además de variantes externas que hacen de nuestro entorno una ventaja más competitiva al aumentar la productividad y satisfacer las necesidades del cliente en el menor tiempo posible para proporcionar mayor valor al negocio. Actualmente existe una gran cantidad de metodologías para el desarrollo de software, separadas en dos grandes grupos; las metodologías tradicionales o pesadas y las metodologías ágiles. Las metodologías ágiles, están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos de desarrollo pequeños, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso [13]. A continuación, se analizan algunas metodologías ágiles tras cumplir con la situación existente.

SCRUM

Su nombre proviene de cierta jugada que tiene lugar durante un partido de rugby, es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo a principios de la década de 1990. En años recientes, Schwaber y Beedle han desarrollado más los métodos de scrum. Sus principios son congruentes con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega. Dentro de cada actividad estructural, las tareas del trabajo ocurren con un patrón del proceso llamado sprint. El trabajo realizado dentro de un sprint se adapta al problema en cuestión y se define y con una frecuencia se modifica en el tiempo real por parte del equipo Scrum. Acentúa el uso de un conjunto de patrones de procesos de software que han demostrado ser eficaces para proyectos con plazos de entrega muy apretados, requerimientos cambiantes y negocios críticos [14]. Como todas las metodologías Scrum cuenta con sus ventajas entre ellas podemos nombrar la gestión de las expectativas del usuario, resultados anticipados, flexibilidad y adaptación a los contextos y gestión sistemática de riesgos. Por otra parte cuenta con limitaciones en cuanto a su implementación debido a que funciona más que nada con equipos reducidos, requiere una exhaustiva definición de las tareas y sus plazos, además exige que quienes la utilicen cuenten con una alta cualificación o formación.

AUP-UCI

El proceso Unificado Ágil (AUP) variación UCI (AUP-UCI) surge de la necesidad de converger a una única metodología de desarrollo que cubra las particularidades de los distintos centros productivos de la UCI. Al igual que AUP (que es una versión simplificada de RUP), describe de una manera simple y fácil de comprender, la forma de desarrollar software utilizando técnicas ágiles y conceptos que aún se mantienen válidos en RUP.[15]

AUP-UCI define 3 fases (inicio, ejecución y cierre), 8 disciplinas (modelado de negocio, requisitos, análisis y diseño, implementación, pruebas internas, pruebas de liberación, pruebas de aceptación, despliegue) y 11 roles (jefe de proyecto, planificador, analista, arquitecto de información, desarrollador, administrador de la configuración, stakeholder (cliente/proveedor de requisitos), administrador de calidad, probador, arquitecto de software, administrador de base de datos). Los requisitos

funcionales pueden ser encapsulados en historias de usuario, en descripción de requisitos por procesos o en casos de uso.[15]

XP

A fin de ilustrar un proceso ágil con más detalles, daremos un panorama de la programación extrema (XP), el enfoque más utilizado del desarrollo de software ágil. Aunque las primeras actividades con las ideas y los métodos asociados XP ocurrieron a finales de la década de 1980, el trabajo fundamental sobre la materia había sido escrito por Kent Beck. Una variante de XP llamada XP industrial (IXP) se propuso en una época más reciente. IXP mejora la XP, y tiene como objetivo el proceso ágil para ser usado específicamente en organizaciones grandes. Beck define un conjunto de cinco valores que establecen el fundamento para todo trabajo realizado como parte de XP: comunicación, simplicidad, retroalimentación, valentía y respeto. Cada uno de estos valores se usa como un motor para actividades, acciones y tareas específicas de XP. La programación extrema usa un enfoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas [14]. Entre sus ventajas se encuentra el uso de la programación organizada, tiene menor tasa de errores, satisfacción del programador, solución de errores de programas, versiones nuevas e implementa una forma de trabajo donde se adapte fácilmente a las circunstancias. Como desventajas se recomienda emplearla solo en proyectos a corto plazo, altas comisiones en caso de fallar, imposible prever todo antes de programar. Se selecciona esta metodología de desarrollo, pues se utiliza fundamentalmente por la poca cantidad de miembros en este caso y permite desarrollarse mediante diferentes programadores, integrarse y ponerse a prueba en un solo día. También indica trabajar sobre los constantes cambios de las funcionalidades y es la metodología propuesta por el cliente.

1.3.2 Lenguajes utilizados

Para la implementación del lado del cliente

HTML5

El HTML es lo que se utiliza para crear todas las páginas web de internet. HTML5 es una nueva especificación que actualmente se encuentra en fase de borrador y que unifica dos elementos tecnológicos. HTML5 es la evolución por un lado de lo que ha

sido hasta ahora el lenguaje de mercado HTML4 y por otro lado de la API Document Object Model 2 (DOM2). A través de HTML5 vamos a disponer de nuevas API's que van a intentar ayudar a los desarrolladores a generar aplicaciones web mucho más dinámicas y ricas. Se tiene siempre claro el punto de vista de que la mayoría de dichas interfaces de programación se encuentran en fase de borrador por lo que podemos afirmar que no están maduras para su implementación en grandes proyectos o aplicaciones por parte de las organizaciones. HTML no se va a quedar solamente en una simple redefinición de etiquetas de marcado, sino que va mucho más allá.[16]

HTML5 presenta las siguientes ventajas [17]:

- Permite el desarrollo de aplicaciones adaptables a distintas resoluciones, tamaño de pantallas, orientaciones, entre otras.
- Los desarrolladores trabajan en lenguajes que no son nativos.
- Tiene elementos semánticos más concretos.

El lenguaje HTML5 se encarga de desarrollar una representación sobre los contenidos utilizados como texto, fotografías, animaciones para definir la estructura básica de la página web.

CCS3

El nombre hojas de estilo en cascada viene del inglés Cascading Style Sheets, del que toma sus siglas (CSS). Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilos que servirán de estándar para los agentes de usuarios o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "<style>". CSS es un lenguaje utilizado para dar estética a un documento HTML (colores, tamaños de las fuentes, tamaños de elemento. Con CSS podemos establecer diferentes reglas que indicarán como debe presentarse un documento. Podemos indicar propiedades como el color, tamaño de la letra, el tipo de letra, si es negrita, si es itálica, también se puede dar

forma a otras cosas que no sean letras, como colores de fondo de una página, tamaños de un elemento (por ejemplo, el alto y el ancho de una tabla).[18]

Ventajas de uso de CCS3 [17]:

- La inclusión de nuevas propiedades de aspecto gráfico.
- El estilo se puede guardar por separado del contenido.
- Genera sitios más rápidos.

El lenguaje CSS3 se emplea para cambiar el aspecto de la página web, desde las medidas para los márgenes hasta las especificaciones para imágenes y texto.

JavaScript v1.8.5

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.[19]

Ventajas que brinda el uso del lenguajes JavaScript [20]:

- Desarrollo de sitios web del lado del cliente (frontend, en el navegador).
- Desarrollo de todo tipo de aplicaciones gracias a la plataforma Node JS.
- Desarrollo de aplicaciones para dispositivos móviles, híbridas o que compilan a nativo.
- Desarrollos de aplicaciones de escritorio para sistemas Windows, Linux, Mac, pudiendo escribir un código compatible con todas las plataformas.

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo, en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Jquery v3.6.0

Es un framework de JavaScript diseñado para hacer a los desarrolladores la vida mucho más fácil. Es un software libre y de código abierto, por lo que puede ser utilizado gratis y con libertad total.[21]

Ventajas que ofrece el framework Jquery [22]:

- Menor cantidad de líneas de código gracias a su sintaxis fácil.
- Manejo independiente de HTML reduciendo el esfuerzo en mantenimiento y aumentando la productividad.
- Adaptabilidad de plugins existentes.
- Basto soporte en documentación y recursos para el desarrollador.

Su principal objetivo es facilitar el uso de JavaScript en el proyecto, debido a que es uno de sus frameworks.

Bootstrap v5.0

Es más popular Front-End Framework de diseño Responsive de código abierto creado por Mark Otto y Jacob Thornton de Twitter, compuesto por HTML, CSS y JavaScript que sirve como estructura de inicio en la producción de aplicaciones web, simplificando este largo proceso y controlando la parte del Front en los sitios.[23]

Facilita la maquetación de sitios web además de ser compatible con procesadores como Less y SaaS, ofrece las herramientas para que un sitio web se vea bien en toda clase de dispositivo, ahorrando así el trabajo de tener que diseñar un sitio web.[24]

Ventajas de usar este framework [23]:

- Estructura : Es un Framework CSS viene definido un sistema de rejilla que permite la colocación de los elementos que conforman la página de manera sencilla en una hoja de estilo ya configurada.
- Compatibilidad : Las páginas elaboradas con este framework serán compatibles con todos los navegadores de escritorio y móviles existentes.

- Código abierto : Su código es abierto a sus usuarios permitiéndoles un libre acceso y manipulación.
- Responsive : Esta es una de las mayores ventajas que tiene, ya que hace posible que las páginas puedan acoplarse y ser visualizadas desde cualquier dispositivo adaptándose a los diferentes tamaños que tienen sin dañar su apariencia.
- Tiempo : Es una ventaja para el desarrollador ya que le ahorra tiempo y esfuerzo en la realización de nuevas páginas web interactivas y funcionales, claro que cuando no se conoce la herramienta el esfuerzo y el tiempo no es tan beneficioso.

Se utiliza para hacer más fácil el trabajo del Front-End en este caso compuesto por HTML, CSS y JavaScript.

Para la implementación del lado del servidor

Python v3.9.0

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Lenguaje interpretado, dinámico y multiplataforma.[20]

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, fue creado a finales de los ochenta por Guido van Rossum en el Centro para las Matemáticas y la Informática (CWI, Centrum Wiskunde & Informatica), en los Países Bajos, como un sucesor del lenguaje de programación ABC, capaz de manejar excepciones e interactuar con el sistema operativo Amoeba. El nombre del lenguaje proviene de la afición de su creador por los humoristas británicos Monty Python.[20]

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Como es un lenguaje interpretado esto significa que no se necesita compilar el código fuente para poder

ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo. En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones y ventajas que brinda como [20] :

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarla desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.
- Además, Python es gratuito, incluso para propósitos empresariales.

Es lenguaje de programación escogido para la propuesta de solución debido a todas las ventajas y facilidades que nos brinda, su uso en el proyecto facilitará y agilizará la solución del mismo.

Django v3.2.4

Es un framework web de código abierto escrito en Python que permite construir aplicaciones web más rápido y con menos código. En el Django Book, Adrian Holovaty y Jacob Kaplan-Moss, definen al desarrollo web como un acto entretenido y creativo; en el peor, puede ser una molestia repetitiva y frustrante. Permite enfocarte en la parte divertida (el quid de tus aplicaciones web) al mismo tiempo que mitiga el esfuerzo de las partes repetitivas. De esta forma, provee un alto nivel de abstracción de patrones comunes en el desarrollo Web, atajos para tareas frecuentes de programación y convenciones claras de como solucionar problemas. Al mismo tiempo, Django intenta no entrometerse, dejándote trabajar fuera del ámbito del framework según sea necesario.[25]

Su objetivo fundamental es facilitar la creación de sitios web complejos. Pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio “No lo repitas” (DRY, del inglés Don’t Repeat Yourself). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.[25]

Entre sus ventajas podemos encontrar [25]:

- Un mapeador objeto-relacional. ORM propio
- Es un sistema cuya arquitectura es multiplataforma.
- Aplicaciones “enchufables” que pueden instalarse en cualquier página gestionada con Django.
- Una API de base de datos robusta.
- Un sistema incorporado de “vistas genéricas” que ahorra tener que escribir la lógica de ciertas tareas comunes.
- Una sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.
- Un despachador de URLs basado en expresiones regulares.
- Soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.
- Documentación incorporada accesible a través de la aplicación administrativa (incluyendo documentación generada automáticamente de los modelos y las bibliotecas de plantillas añadidas por aplicaciones).
- Consola de administración de proyectos Django.
- Un sistema “middleware” para desarrollar características adicionales; por ejemplo, la distribución principal de Django incluye componentes middleware que proporcionan cacheo, compresión de la salida, normalización de URLs, protección CSRF y soporte de sesiones.

La funcionalidad de este framework en el proyecto es facilitar el trabajo con Python, además de disfrutar de todas las ventajas que nos brinda y nos permitirá una solución en corto plazo.

1.3.3 Entorno de desarrollo

Uno de los factores importantes a tener en cuenta, a la hora de abordar el desarrollo de software, es el conjunto de herramientas con el que podemos contar para realizar

el trabajo. Con independencia de la tecnología y el lenguaje, existen diferentes tipos de herramientas de desarrollo de software, desde un sencillo editor de texto, hasta complejos depuradores, pasando por entornos integrados de desarrollo que ofrecen bastantes funcionalidades en un solo programa. Python no es una excepción y cuenta con diferentes herramientas de desarrollo que nos ayudarán a ser más productivos.[26]

Podemos considerar a los editores de texto como las herramientas básicas para desarrollar software, ya que nos permiten escribir el código fuente y crear un fichero a partir del mismo.[26]

La evolución de los editores de código son los entornos integrados de desarrollo por sus siglas en inglés (IDE). Estos amplían la funcionalidad de los editores añadiendo facilidades para la depuración de código, la creación de proyectos, el auto completado, la búsqueda de referencias en la documentación o el marcado de sintaxis errónea.[26]

Visual Studio Code

Es un software gratuito que cualquier usuario puede descargar, tiene un soporte de 64 bits, soporta una cantidad considerable de lenguajes ya sean propios de Microsoft como C#, F#, Visual Basic, PHP, Python, Perl, SQL, Java, .NET, HTML, CSS y JavaScript.[27]

Visual Studio Code es un editor muy ligero, multiplataforma y muy modular (Macos, Linux y Windows) que ofrece ventajas como[27]:

- Interfaz moderna que permite al usuario acoplarse desde el primer instante que empiece a usarlo.
- Brinda la opción de editor de código, esta opción genera una ventaja porque cada línea de código se presenta con colores.
- Completo depurador tiene la opción que brinda verificar las distintas variables, y generar una consola con sus comandos respectivos.
- Adjunto de Git : genera al usuario poder controlar las distintas actualizaciones sin tener que realizar ninguna edición.

- **Compatibilidad** : este programa cuenta con la opción de manejar varios lenguajes de programación.
- **Manejable** : permite que el usuario lo personalice de la manera más conveniente con él, ya que brinda la opción de ordenarlo por bloques.

PyCharm v2019.2

Es el IDE más popular para desarrollar Python, incluye excelentes características, terminación e inspección de código con depurador avanzado y soporte para la programación web y varios marcos. Creado por la compañía Checa, JetBrains, que se enfoca en crear un entorno de desarrollo integrado para varios lenguajes de desarrollo web, como JavaScript y PHP [24]. Es una software multiplataforma, puede ser utilizado en Windows, macOS y Linux, es utilizado especialmente en el lenguaje Python.

Ventajas de PyCharm [28] :

- Permite completar código con algunos atajos de teclado. Así mismo, permite navegar a través de nuestro código, saltando entre las clases y métodos creados, haciendo el flujo de trabajo mucho más dinámico.
- Una de las características notables, es la posibilidad que tiene de Re factorizar el código, que en términos generales, significa modificar el código sin comprometer la ejecución del mismo.
- Esta operación se realiza de forma constante dentro de la Ingeniería de Software y es más conocida como limpiar el código para que este pueda ser interpretado con facilidad cuando hay distintas personas integrando un equipo de trabajo.
- Por último, la gran cantidad de desarrolladores que trabajan con PyCharm ha generado que se tenga una gran cantidad de temas y plugins que se pueden usar para trabajar más cómodamente.

Se escoge como entorno de desarrollo para este proyecto, porque es uno de los más completo para trabajar con Python y también uno de los más populares, además permite el trabajo con HTML5, CSS, JavaScript, JQuery y Bootstrap. Una de las ventajas importante es la de refactorizar el código, es decir, simplificar el código

sin cambiar su función para que pueda ser entendido con mayor facilidad. Por otra parte cuenta con un editor inteligente en el que se puede utilizar una gran cantidad de atajos de teclado.

1.3.4 Sistemas gestores de base datos

Los sistemas gestores de bases de datos, en adelante SGBD, aparecen cómo una capa intermedia entre los programas que el usuario final utiliza y el sistema operativo. Los SGBD son, por tanto el sistema encargado de establecer la comunicación entre estos dos sistemas. Hemos visto ya que antes de aparecer las bases de datos y los SGBD, las aplicaciones eran las encargadas de definir su propio sistema de ficheros para almacenar datos de forma que, cada vez que cambiara la aplicación, era necesario cambiar la estructura de todos los ficheros y al contrario. Con el nacimiento de los SGBD cómo un sistema intermedio entre las bases de datos y las aplicaciones, se evitan todos estos inconvenientes.[29]

El SGBD es, una aplicación que permite a los usuarios definir, crear y mantener la BD y proporciona un acceso controlado a la misma. Un SGBD debe prestar los siguientes servicios [29]:

- Definición y creación de la base de datos.
- Manipulación de los datos realizando consultas, inserciones y actualizaciones.
- Acceso controlado a los datos mediante mecanismos de seguridad de acceso a los usuarios.
- Mantener la integridad de los datos.
- Controlar la concurrencia a la base de datos.
- Mecanismos de copias de respaldo y recuperación para restablecer la información en caso de fallos en el sistema.

MySQL: Es una base de datos de código abierto. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de MySQL. Es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización.[30]

MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcados de datos. Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos, utiliza el lenguaje de consulta

estructurado (SQL). Se trata del lenguaje utilizado por todas las bases relacionales, este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos.[30]

MongoDB: Es un sistema de base de datos NoSQL, multiplataforma libre que es orientada a documentos quiere decir que se agregan los datos o información que deseemos en documentos en vez de que se almacene en registros como es el caso de las bases de datos SQL, con un conjunto de documentos se forma una colección de documentos que en este caso serían las tablas lo cual nos brinda un valioso rendimiento, disponibilidad y escalabilidad.[31]

Es un sistema gestor de base de datos muy estable para el almacenamiento de información, el control de acceso y la administración entre roles para usuarios que se caracteriza principalmente en la manipulación de la información a través de código en JavaScript, además de su gran velocidad de consulta como lo probamos anteriormente ya que los datos se guardan en formato JSON, y que contiene una escalabilidad horizontal para aumentar moderadamente la capacidad de trabajo y tamaño.[31]

PostgreSQL: Es un sistema de base de datos relacional y libre, que agiliza la interacción de cliente, servidor y base de datos, donde PostgreSQL es el que realiza la mayoría del trabajo referente a bases de datos cuando se le hacen peticiones. También es un SGBD orientado a objetos, usado en entornos de software libre, porque cumple con todos los estándares (SQL92 y SQL99) y también un conjunto de funcionalidades avanzadas que soporta, fue desarrollada en la universidad de Berkeley, y se distribuye bajo la licencia BSD, es el sistema de gestión de bases de datos de código abierto más potente del mercado.[31]

Sistema estable de almacenamiento, control, como lo es PgAdmin, que se caracteriza principalmente que la mayoría de gestores de bases de datos ya que se lo puede instalar y configurar en sistemas operativos tanto libres como privativos. Permite la administración, componer nuevos sistemas y aplicaciones con buen manejo de este gestor. [31]

Este gestor de bases de datos permite migrar y desarrollar aplicaciones desde Visual Basic, Access, Visual C/C++, entre otras. Es una gran alternativa de uso, este gestor es un sistema estable, con gran flexibilidad. PostgreSQL está agregando características acerca de las bases de datos NoSQL, como lo es el tipo de archivo JSON esta característica se le está añadiendo por el uso paulatino por parte de MongoDB.[31]

Se escoge como sistema gestor de base de datos el PostgreSQL v13. Permite que los fallos no afecten al resto del sistema. Además el PgAdmin4 v5.0 brinda la facilidad para hacer mantenimiento de las tablas o respaldos, Host-Standby permite que los usuarios puedan acceder a las tablas en modo lectura mientras que se realizan los procesos de backup o mantenimiento. Es importante tener en cuenta varios factores como la performance y la fiabilidad, con PostgreSQL tenemos una base datos relacional que nos garantiza la consistencia de los datos, con MongoDB una base datos no relacional con una menor latencia. Por otra parte MySQL resulta ser un sistema en general más simple, utiliza bases de datos más pequeñas o medianas, de lo contrario PostgreSQL consta con más funciones y trabaja con bases de datos muy grandes.

1.3.5 Herramientas CASE

Las herramientas CASE son aplicaciones informáticas destinadas a aumentar la productividad y la calidad en el desarrollo del software reduciendo los costos del mismo en términos de tiempo y dinero. Estas herramientas permiten tener una mejor organización y control del desarrollo de un sistema informático, en especial aquellos sistemas que sean grandes o robustos y que impliquen tener muchos componentes software así como recursos humanos.[32]

En la actualidad muchas instituciones utilizan herramientas CASE para automatizar las actividades involucradas en el proceso de desarrollo de los sistemas informáticos y así obtener grandes beneficios, tales como mejorar la gestión y dominio sobre el proyecto (planificación, ejecución y control); una mejor calidad del software desarrollado así como también aumentar la productividad en las áreas de desarrollo y mantenimiento de dichos sistemas (ejemplo, estandarizar la documentación, facilitar la realización de prototipos y la reutilización de componentes software) con la finalidad de ser competitivo en el mercado.[32]

Visual Paradigm v16.2

Visual Paradigm es un galardonado paquete de gestión empresarial y desarrollo de software, que proporciona todas las herramientas necesarias para la arquitectura empresarial, la gestión de proyectos, el desarrollo de software y la colaboración en equipo. Diseñado para una amplia gama de usuarios, incluidos arquitectos empresariales, gerentes de proyectos, desarrolladores de software, analistas de

sistema, analistas de negocios, arquitectos de sistemas y cualquier persona que necesite construir sistemas de software a gran escala.[33]

Propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y diseño, hasta la generación de código fuente de los programas y documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Esta herramienta permite aumentar la calidad del software, mediante la mejora de la productividad en el desarrollo y mantenimiento del software. Aumenta el conocimiento informático de una empresa ayudando así a la búsqueda de soluciones para los requisitos. También permite la reutilización del software, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería de Software.[34]

1.3.6 Conclusiones Parciales

En el presente capítulo se abordaron elementos referentes al caso de estudio en cuestión, en tal sentido se puede arribar a las siguientes conclusiones:

- La definición de conceptos relacionados con la seguridad y salud en el trabajo permitió una mayor comprensión de el funcionamiento de los SG-SST.
- El análisis de sistemas homólogos, permitió identificar las tendencias en el desarrollo de herramientas informáticas para la gestión de la seguridad y salud en el trabajo, así como comprender la necesidad de desarrollar un nuevo Sistema de Gestión de Accidentes y Peligros para el Control de la seguridad y salud en el trabajo debido a que ninguno de los sistemas señalados reúne los requisitos necesarios para ser utilizado directamente como propuesta de solución.
- Se adopta como metodología de desarrollo XP, se describen los lenguajes a utilizar y se selecciona como lenguaje de programación Python y como framework Django, además se analizan las herramientas de desarrollo y se escoge como entorno de desarrollo PyCharm v2019.2, como sistema gestor de base de datos PostgreSQL v13 y herramienta CASE se utilizará Visual Paradigm v16.2.

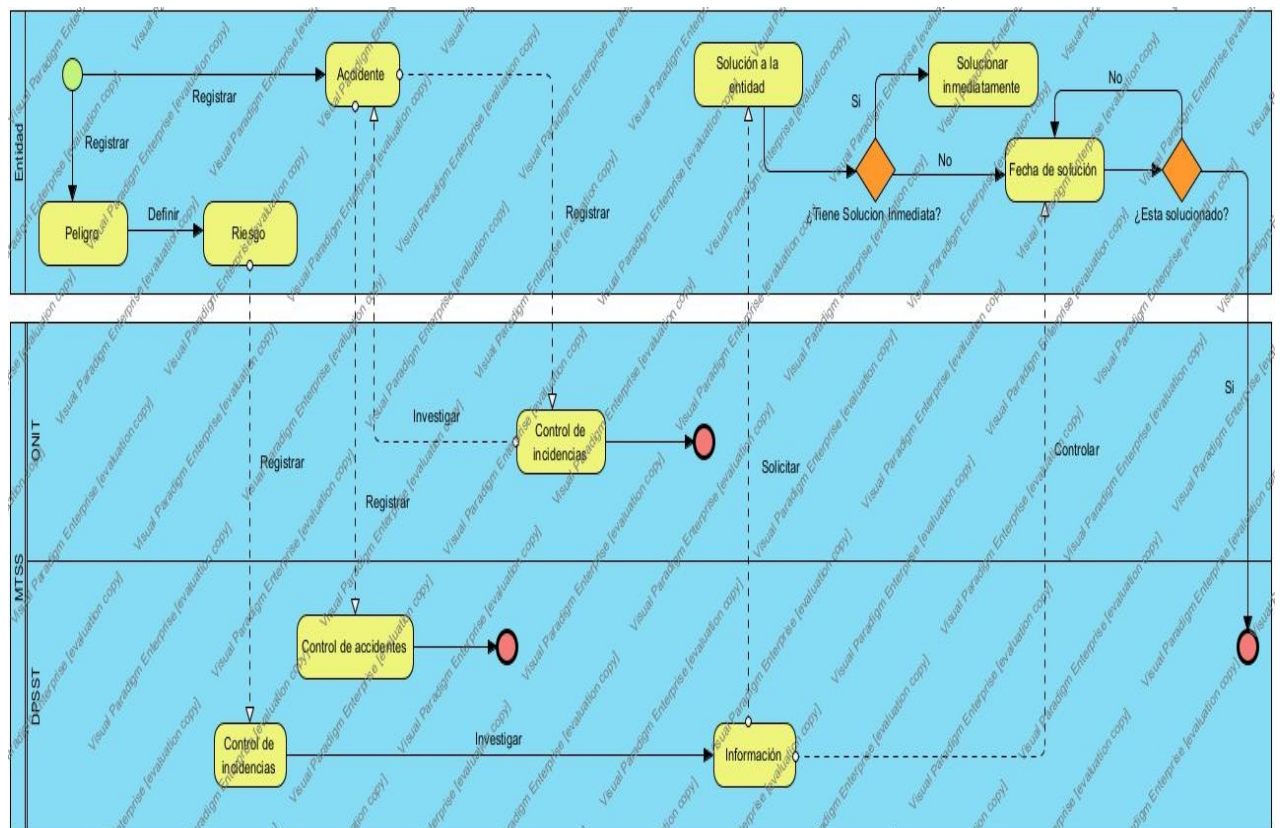
Capítulo II. Análisis y diseño del sistema

En el presente capítulo se realiza un análisis del proceso del negocio a través de un modelo de proceso de negocio. Además, se presentan los distintos requisitos funcionales y no funcionales que presenta el sistema. Por otra parte, se estructura mediante las fases definidas por XP planeación, diseño, codificación y prueba, para lograr el entendimiento del funcionamiento del sistema se describe a través de las historias de usuarios sus distintas funcionalidades.

2.1 Modelo de proceso de negocio

Un modelo es una fotografía de cómo los modeladores ven el negocio en un momento determinado y posibilita hacer cambios y proponer mejoras. Facilita el entendimiento humano y la comunicación, además la gestión, el control y la mejora de procesos, siendo también una base para su definición y análisis. El modelo revela lo que es importante mostrar, desde el punto de vista de la persona que lo crea, para entender o predecir la situación modelada.[35]

Figura 1. Diagrama de proceso de negocio



2.1.1 Descripción del proceso de negocio

El proceso de la gestión de peligros y accidentes para el control de la seguridad y salud en el trabajo: Surge de la necesidad de controlar y disminuir los accidentes laborales y a su vez detectar peligros para evitar que ocurran accidentes en las diferentes entidades del país.

Se desglosa en 4 niveles describiendo un orden por el cual debe pasar el proceso de gestión de accidentes y peligros. Dichos niveles son:

1. Entidad (es donde surgen los accidentes y peligros)
2. Oficina Nacional de Inspección del Trabajo (ONIT), (en caso que sea accidente)
3. Departamento de Seguridad y Salud en el Trabajo (DPSST), (en ambos casos)
4. Ministerio de Trabajo y Seguridad Social (MTSS), (en ambos casos)

El proceso se divide en dos partes, primero la Entidad registra el peligro detectado, luego define el riesgo que este puede ocasionar lo registra en el control de incidencias del DPSST perteneciente al MTSS, después este investiga el peligro, da la información y solicita a la Entidad una solución. Si el peligro detectado requiere solución inmediata, la Entidad tiene que dársela, si no se definirá una fecha de solución, el DPSST se encarga de controlar que se cumpla con dicha fecha de solución y si está solucionado el problema, concluye el proceso, si no propone una nueva fecha así hasta que se soluciones este peligro.

La segunda parte de este proceso se lleva a cabo cuando ocurre un accidente, la Entidad se encarga de registrarlo y hacerlo llegar al control de incidencias de la ONIT y al DPSST, donde luego se lleva a cabo un investigación, para posteriormente llegar a los responsables y dar una conclusión para así terminar el proceso.

2.2 Roles y responsabilidades

A continuación se muestran los roles y sus responsabilidades correspondiendo al sistema que se implementará:

- 1) **Administrador:** Es el responsable de mantener la disponibilidad de los datos y la correcta gestión del proceso, además de darle cierre a un evento y administrar en el sistema, los niveles de acceso de los usuarios.
- 2) **Entidad:** Es el encargado de gestionar la información de los peligros y accidentes laborales dentro de la entidad.
- 3) **Organismo:** Visualiza la información de todas las entidades que se subordinan a él.
- 4) **DMT:** Encargado de visualizar toda la información de las entidades que se encuentran en los municipios de las DMT.
- 5) **DPT:** Encargado de visualizar toda la información de las entidades que se encuentran en los municipios de las DPT.
- 6) **Especialista:** Visualiza y gestiona toda la información a nivel de país.

2.3 Fase de Planeación

La actividad comienza desarrollando requerimientos que permite que los miembros técnicos que emplea la metodología XP entienda el contexto del negocio para el software y adquieran la sensibilidad de la salida, características principales y

funcionalidades que se requieren. Esto lleva a la creación de algunas historias del usuario que describen la salida necesaria, características y funcionalidad del software que se va a elaborar.[14]

Cada historia es escrita por el cliente y colocada en un tarjeta indizada. El cual asigna un valor a la historia con base en el valor general de la característica o función para el negocio. Posteriormente, los miembros del equipo XP evalúan cada historia y le asignan un costo, medido en semanas de desarrollo. Una vez llegado a un compromiso sobre la entrega (acuerdo sobre las historias por incluir, la fecha de entrega y otros aspectos del proyecto), el equipo que emplea la metodología XP ordena las historias que serán desarrolladas, medida que avanza el trabajo, el cliente puede agregar historias, cambiar el valor de una ya existente, descomponerlas o eliminarlas.[14]

2.3.1 Levantamiento de Requisitos

El proceso de captura de requisitos inicia con la interacción con el cliente. Durante este proceso se hace una relación de las necesidades que tiene el usuario y se define lo que debe hacer el sistema. Se detallan los usuarios que van a interactuar con el sistema y los niveles de acceso o permiso que tendrán que tendrán en el. Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación.[15]

Requisitos funcionales (RF)

En XP se elimina la fase inicial de captura de requisitos y se permite que estos se vayan definiendo de una forma ordenada durante el tiempo que dura el proyecto. El cliente puede cambiar de opinión sobre la marcha y a cambio debe encontrarse siempre disponible para resolver dudas del equipo de desarrollo y para detallar los requisitos especificados cuando sea necesario. El proceso de captura de requisitos de XP gira entorno a una lista de características que el cliente desea que existan en el sistema final [36]. A continuación, se listan las funcionalidades del sistema. En la tabla 3 se identifican los requisitos funcionales.

Tabla 3. Casos de usos que responden requisitos funcionales

No.	Requisitos Funcionales	No.	Requisitos Funcionales
RF1	Registrar Usuario	RF52	Modificar Organismo
RF2	Autenticar Usuario	RF53	Eliminar Organismo
RF3	Cerrar Sesión	RF54	Deshabilitar Organismo
RF4	Listar Usuario	RF55	Habilitar Organismo

RF5	Modificar Usuario	RF56	Eliminar Organismos Seleccionados
RF6	Detalles de Usuario	RF57	Buscar Organismo
RF7	Eliminar Usuario	RF58	Registrar Accidente laboral
RF8	Eliminar Usuarios Seleccionados	RF59	Listar Accidentes laborales
RF9	Deshabilitar Usuario	RF60	Modificar Accidente laboral
RF10	Habilitar Usuario	RF61	Detalles del Accidente laboral
RF11	Cambiar contraseña de Usuario	RF62	Eliminar Accidente laboral
RF12	Registrar Categorías de Usuario	RF63	Eliminar Accidentes laborales Seleccionados
RF13	Listar Categorías de Usuario	RF64	Habilitar Accidente laboral
RF14	Modificar Categoría de Usuario	RF65	Deshabilitar Accidente laboral
RF15	Eliminar Categoría de Usuario	RF66	Buscar Accidente laboral
RF16	Deshabilitar Categoría de Usuario	RF67	Registrar Persona
RF17	Habilitar Categoría de Usuario	RF68	Listar Personas
RF18	Eliminar Categorías de Usuario Seleccionadas	RF69	Modificar Persona
RF19	Registrar Provincia	RF70	Eliminar Persona
RF20	Listar Provincias	RF71	Habilitar Persona

RF21	Modificar Provincia	RF72	Deshabilitar Persona
RF22	Eliminar Provincia	RF73	Eliminar Personas Seleccionadas
RF23	Deshabilitar Provincia	RF74	Detalles de la Persona
RF24	Habilitar Provincia	RF75	Buscar Persona
RF25	Eliminar Provincias Seleccionadas	RF76	Registrar Peligro
RF26	Registrar Municipio	RF77	Listar Peligros
RF27	Listar Municipios	RF78	Modificar Peligro
RF28	Modificar Municipio	RF79	Eliminar Peligro
RF29	Eliminar Municipio	RF80	Habilitar Peligro
RF30	Buscar Municipio	RF81	Deshabilitar Peligro
RF31	Deshabilitar Municipio	RF82	Buscar Peligro
RF32	Habilitar Municipio	RF83	Eliminar Peligros Seleccionados
RF33	Eliminar Municipios Seleccionados	RF84	Detalles del Peligro
RF34	Registrar Entidad	RF85	Registrar Riesgo
RF35	Listar Entidades	RF86	Listar Riesgos
RF36	Modificar Entidad	RF87	Modificar Riesgo
RF37	Eliminar Entidad	RF88	Eliminar Riesgo
RF38	Deshabilitar Entidad	RF89	Habilitar Riesgo
RF39	Habilitar Entidad	RF90	Deshabilitar Riesgo
RF40	Eliminar Entidades Seleccionadas	RF91	Eliminar Riesgos Seleccionados
RF41	Buscar Entidad	RF92	Detalles del Riesgo
RF42	Registrar Sector	RF93	Buscar Riesgo
RF43	Listar Sectores	RF94	Exportar Reportes
RF44	Modificar Sector		
RF45	Eliminar Sector		

RF46	Deshabilitar Sector		
RF47	Habilitar Sector		
RF48	Eliminar Sectores Seleccionados		
RF49	Buscar Sector		
RF50	Registrar Organismo		
RF51	Listar Organismo		

Requisitos no funcionales (RNF)

Los requerimientos no funcionales (nonfunctional requirements o NFR en inglés), representan cualidades generales y restricciones que afectan a aplicaciones y sistemas enteros. A diferencia de los epics e historias de usuario, cuyos criterios de aceptación se centran en comportamientos y funciones específicas. Sin embargo una historia o epic no se puede considerar hecha si no cumple con los requisitos no funcionales asociados. Estos garantizan que el sistema cumple con cualidades de gran importancia como usabilidad, seguridad, software, portabilidad, implementación, accesibilidad, entre otros.[37]

Tabla 4. Requisitos no funcionales

No.	Requisitos no funcionales del sistema.
Usabilidad	
RNF1	El sistema debe presentar un acceso fácil e interfaz amigable con informaciones fáciles de entender, para facilitar el uso del mismo por usuarios con pocos conocimientos en el campo de la informática.
Seguridad	
RNF2	Se asignarán los permisos de acceso en dependencia del rol que desempeñe cada usuario del sistema.
RNF3	Se debe contar con un sistema de salvadas externas para la información que maneja el sistema.
Software	
RNF4	Intérprete de aplicaciones Web (Mozilla Firefox 25.0, Internet Explorer 10, Google Chrome 32.0.1700.107).

RNF5	Sistema Gestor de Base Datos PostgreSQL.
Portabilidad	
RNF6	El sistema será multiplataforma, basado en sistemas operativos GNU/Linux y/o Windows.
Implementación	
RNF7	Hojas de Estilos en Cascada 3 (CSS).
RNF8	SQL como lenguaje de base de datos.

2.3.2 Historias de usuario (HU)

Las historias de usuario se usan, en el contexto de la ingeniería de requisitos ágil, como una herramienta de comunicación que combina las fortalezas de ambos medios: escrito y verbal. Describen, en una o dos frases, una funcionalidad del software desde el punto de vista del usuario, con el lenguaje que éste emplearía. El foco está puesto en qué necesidades o problemas soluciona lo que se va construir. Su origen viene de la metodología XP, donde deben ser escritas por los clientes.[37]

Las Historias de usuario son una herramienta que agiliza la administración de requisitos, reduciendo la cantidad de documentos formales y tiempo necesarios. Forman parte de la fórmula de captura de funcionalidades definida en 2001 por Ron Jeffries de las tres Cs: (Card, Conversation y Confirmation) [37]. Las HU restantes se encuentran en **Anexo I Historias de Usuario**.

Tabla 5. Ejemplo de Historia de usuario: Autenticar usuario

Historia de usuario	
Número: HU_1	Nombre historia de usuario: Autenticar usuario
Modificación de historia de usuario: 2	
Programador: Harold Pérez Trujillo	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 7 días

Riesgo en Desarrollo: Alta	Tiempo Real: 7
Descripción: muestra los campos correspondientes a cada tipo de contenido, los que pueden ser seleccionados por los usuarios a autenticar.	
Observaciones: el campo contenido está seleccionado por defecto.	

La prioridad:

Alta: Son aquellas HU que constituyen funcionalidades fundamentales en el desarrollo el sistema, a las que el cliente define como principales para el control integral del sistema.

Media: Son las funcionalidades a tener en cuenta por el cliente, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.

Baja: Es otorgada a las HU que son funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo.

Leyenda de las Historias de Usuario:

Número: número de identificación para las HU, sería incremental en el tiempo.

Nombre historia de usuario: es el nombre de la HU, sirve para identificarla fácilmente tanto para los desarrolladores como para los clientes.

Modificación de historia de usuario: cantidad de modificaciones que se le ha realizado a la HU (de no tener modificaciones se pone ninguna, si no la cantidad de veces que ha sido modificada).

Programador: nombre del programador encargado de implementar la HU.

Prioridad: qué tan importante es para el cliente, se clasifica en Alta, Media y Baja.

Iteración Asignada: iteración en la que se desarrollará la HU.

Riesgo en Desarrollo: riesgo que puede correr el desarrollo de está, se clasifica en Alta, Media y Baja.

Tiempo Estimado: tiempo en días que se le asignará.

Tiempo Real: tiempo que tomó en culminarla.

Descripción: es la descripción de la historia, detallando las operaciones del usuario y las respuestas del sistema.

Observaciones: informaciones de interés, como glosarios, detalles del usuario, entre otros.

2.4 Fase de Diseño

La metodología XP estimula el uso de las tarjetas CRC como un mecanismo eficaz para pensar en el software en un contexto orientado a objetos. Las tarjetas CRC (clase-responsabilidad-colaborador) identifican y organizan las clases orientadas a objetos que son relevantes para el incremento actual del software. Son el único producto del trabajo de diseño que se genera como parte del proceso XP. Si en el diseño de una historia se encuentra un problema de diseño difícil, XP recomienda la creación inmediata de un prototipo operativo de esa porción del diseño. Entonces, se implementa y evalúa el prototipo del diseño, llamado solución en punta. El objetivo es disminuir el riesgo cuando comience la implementación verdadera y validar las estimaciones originales para la historia que contiene el problema de diseño.[14]

2.4.1 Tarjetas CRC

Como parte de la fase de diseño en la metodología XP es el uso de las tarjetas CRC como un mecanismo eficaz para pensar en el software en un contexto orientado a objetos. Son el único producto del trabajo de diseño que se genera como parte del proceso XP, se dividen en tres secciones. En la parte superior de la tarjeta se escribe el nombre de la clase, en la parte izquierda se listan las responsabilidades de la clase y en la derecha, los colaboradores. Las tarjetas CRC restantes se encuentran en el **Anexo II Tarjetas CRC**.

Tabla 6. Ejemplo de Tarjeta CRC: Gestionar Usuario

Nombre de la clase: Usuario	
Responsabilidades de la clase: -Registrar un nuevo usuario. -Iniciar sesión o autenticarse. -Cerrar sesión. -Listar todos los usuarios registrados. -Modificar o editar un usuario. -Mostrar detalles del usuario. -Eliminar un usuario determinado. -Eliminar usuarios seleccionados. -Habilitar o deshabilitar un usuario. -Cambiar contraseña del usuario.	Colaboradores:

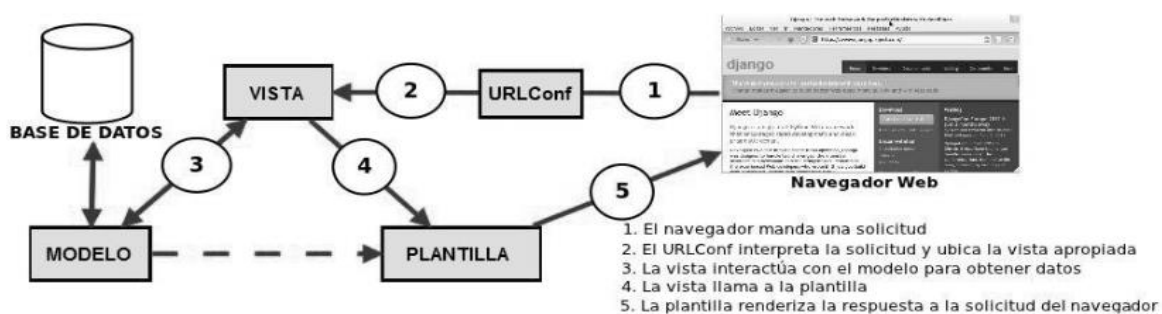
2.4.2 Arquitectura de software

La arquitectura del software alude a la estructura general de éste y a las formas en las que ésta da integridad conceptual a un sistema. En su forma más sencilla, la arquitectura es la estructura de la organización de los componentes de un programa (módulos), la forma en la que éstos interactúan y la estructura de datos que utilizan. Sin embargo, en un sentido más amplio, los componentes se generalizan para que representen los elementos de un sistema grande y sus interacciones. Una meta del diseño del software es obtener una aproximación arquitectónica de un sistema. Ésta sirve como estructura a partir de la cual se realizan las actividades de diseño más detalladas. Un conjunto de patrones arquitectónicos permite que el ingeniero de software resuelva problemas de diseños comunes.[14]

Django usa una modificación de la arquitectura Modelo-Vista-Controlador (MVC), llamada MTV (Model - Template - View), que sería Modelo-Plantilla-Vista, esta forma de trabajar permite que sea pragmático. Esto se debe a que los desarrolladores no tuvieron la intención de seguir algún patrón de desarrollo, sino hace el framework lo más funcional posible [38]. Esta arquitectura (MTV) constituye una adaptación de la mencionada anteriormente (MVC), que en esencia realiza algunos cambios de

conceptos en las responsabilidades de las distintas capas. La capa del modelo, sigue teniendo la misma función de acceso a datos; la capa Vista, a su función en MVC, es responsable de aislar toda la lógica del negocio. Esta capa sirve de “puente” entre el modelo y la capa de presentación o interfaz de usuario llamada ahora Plantilla, que por lo general la constituyen archivos de código HTML y que se conecta con la Vista siempre a través de una URL [39]. En la siguiente figura se expone el funcionamiento del patrón arquitectónico MVT para una mayor comprensión.

Figura 2. Patrón arquitectónico MVT [40].

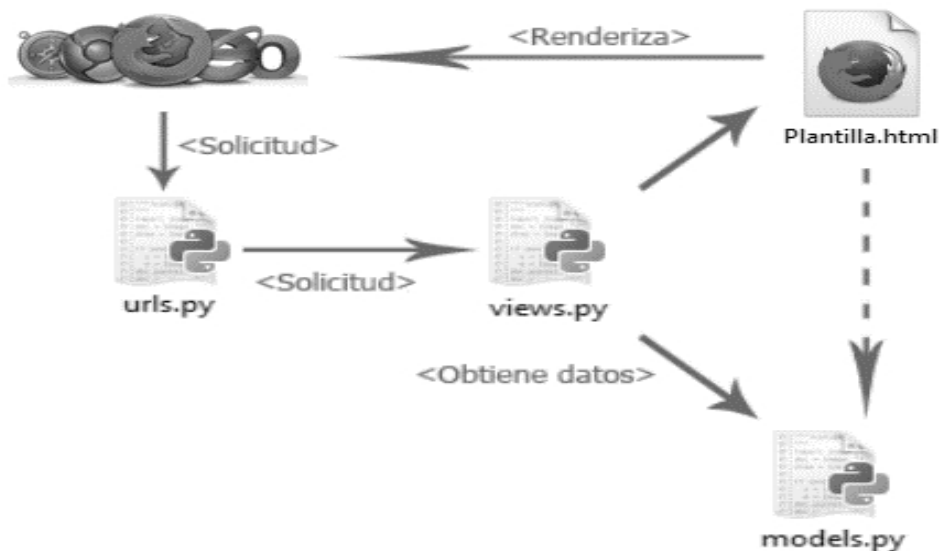


A partir de lo antes expuesto se derivan los siguientes elementos [41]:

- El **modelo** define los datos almacenados, se encuentra en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros, posee métodos también. Todo esto permite indicar y controlar el comportamiento de los datos.
- La **vista** se presenta en formas de funciones o clases en Python, su propósito principal es determinar qué datos serán visualizados. EL ORM de Django permite escribir código Python en lugar de SQL para hacer las consultas que necesita la vista. La vista también se encarga de tareas conocidas como el envío de correo electrónico, la autenticación con servicios externos, entre otras. Lo más importante a entender con respecto a la vista es que no tiene nada que ver con el estilo de la presentación de los datos, solo se encarga de los datos, la presentación es tarea de la plantilla.
- La **plantilla** es básicamente una página HTML con algunas etiquetas extras propias de Django, en sí no solamente crea contenido en HTML (también XML, CSS, Javascript, CSV, entre otros).

En general, cada una de las aplicaciones del sistema cuenta con un fichero “models.py” donde se definen todos los modelos, “views.py” para definir los controladores y una serie de archivos HTML que representan las vistas. Para una mejor comprensión de lo antes planteado, se muestra un ejemplo del patrón en la implementación del sistema:

Figura 3. Ejemplo del patrón MVP utilizado en la implementación del sistema

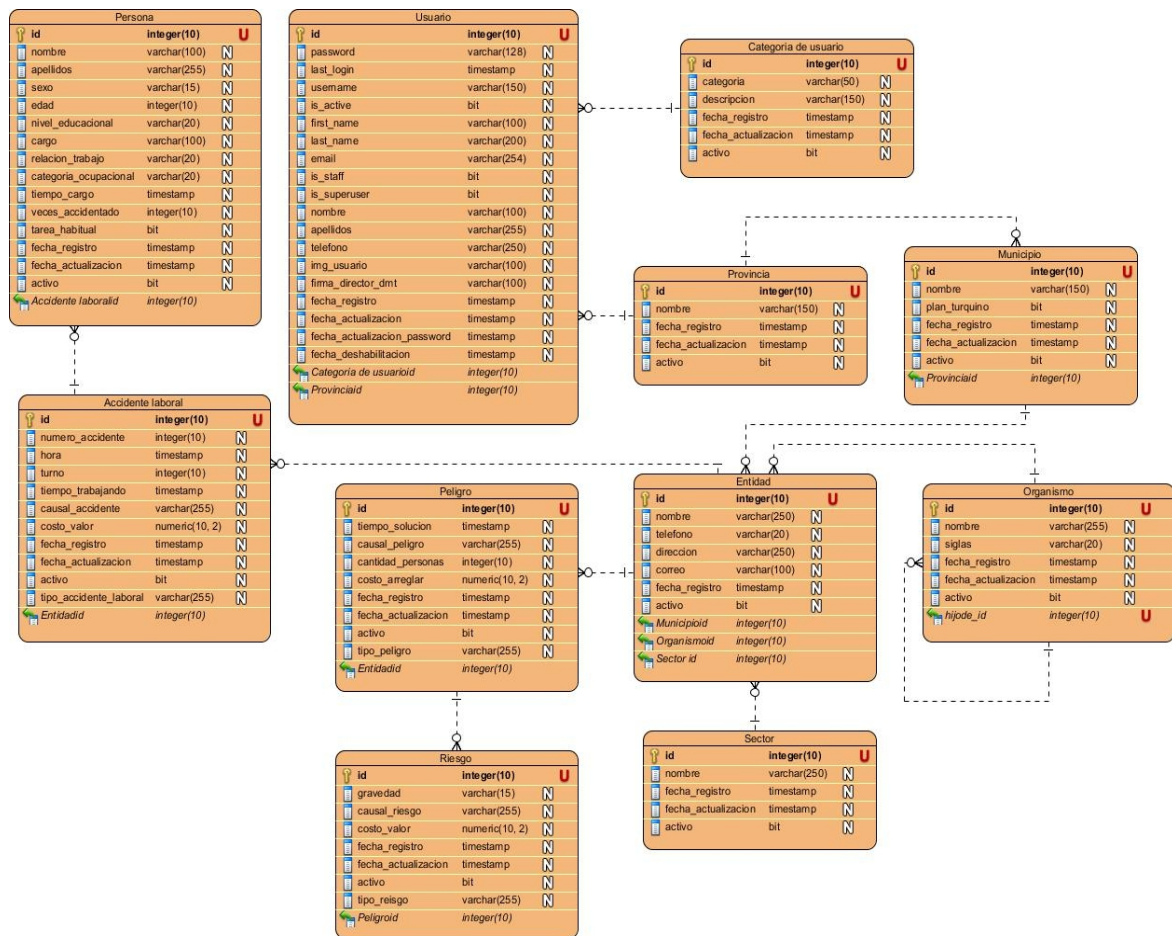


2.5 Modelo de datos

Si los requerimientos del software incluyen la necesidad de crear, ampliar o hacer interfaz con una base de datos, o si deben construirse y manipularse estructuras de datos complejas, el equipo del software tal vez elija crear un modelo de datos como parte del modelado general de los requerimientos. Un ingeniero o analista de software define todos los objetos de datos que se procesan dentro del sistema, la relación entre ellos y otro tipo de información que sea pertinente para las relaciones. El diagrama Entidad-Relación (DER) aborda dichos aspectos y representa todos los datos que se introducen, almacenan, transforman y generan dentro de una aplicación.[14]

A continuación, la imagen muestra el diagrama Entidad-Relación que modela la base de datos de la aplicación:

Figura 4. Modelo de datos



2.6 Patrones de diseño.

Los patrones de diseños se derivaron de ideas planteadas por Christopher Alexander, quien sugirió que había ciertos patrones comunes de diseño de construcción que eran relativamente agradables y efectivos. El patrón es una descripción del problema y la esencia de su solución, de modo que la solución puede reutilizarse en diferentes configuraciones. El patrón no es una especificación detallada. Más bien, puede considerarla como una descripción de sabiduría y experiencias acumuladas, una solución bien probada a un problema común.[42]

En el diseño de la solución propuesta se hace uso de los siguientes patrones:

Patrones GRASP (Responsability Assignment Software Patterns): Patrones generales de software para asignar responsabilidades a objetos.

- **Creador:** Este patrón como su nombre lo indica es el que crea, el que guía la asignación de responsabilidades relacionadas con la creación de objetos, se asigna la responsabilidad de que una clase B cree un objeto de la clase A.

Este patrón se manifiesta en el siguiente fragmento de código, al crear un objeto como instancia de la clase `Accidentelaboral` sobre las cuales se realizan las operaciones necesarias para eliminarla:

Figura 5. Código para eliminar un determinado Accidente laboral

```
class EliminarAccidentelaboralView(PermissionRequiredMixin, View):
    permission_required = ['administrador']

    def get(self, request, *args, **kwargs):
        accidente = Accidentelaboral.objects.get(id=self.kwargs['pk'])
        accidente.delete()
        messages.add_message(request, messages.SUCCESS, "Accidente Laboral eliminado con éxito.")
        return JsonResponse({})
```

- **Controlador:** Se asocia con operaciones del sistema y respuestas a sus eventos, tal como se relacionan los mensajes y los métodos. El controlador delega en otros objetos el trabajo que se necesita hacer, pero coordina o controla la actividad.

El patrón controlador es utilizado en el siguiente fragmento de código, donde se llama a la función “`objects.all()`” para obtener todos los `Accidentelaboral` de la base de datos y confeccionar un listado con estos elementos

Figura 6. Código para listar los Accidentes laborales

```
class ListadoAccidentelaboralView(PermissionRequiredMixin, ListView):
    model = Accidentelaboral
    template_name = 'accidente_laboral/listado_accidentes_laborales.html'
    paginate_by = 10
    permission_required = ['administrador', 'especialista']

    def get_queryset(self):
        accidentes = Accidentelaboral.objects.all().order_by('id')

        q = self.request.GET.get('q')
        if q is not None and q != "":
            accidentes = accidentes.filter(nombre__icontains=q)

        return accidentes

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['titulo'] = 'Listado de accidentes laborales'
        context['activar_accidentes_laborales'] = True
        context['path'] = [
            {'name': 'Accidentes laborales', 'href': reverse_lazy('accidentes_laborales')}
        ]
        return context
```

Patrones GoF (Gang of Four): Representan soluciones técnicas basadas en POO que favorecen la reutilización del código.

- **Decorador:** El patrón Decorador es usado cuando es necesario adicionar, en tiempo de ejecución más funcionalidades a un objeto de una clase. En el caso de Python se utiliza las denominadas funciones de segundo orden provenientes de la programación funcional, las cuales reciben un objeto por parámetro y devuelven el mismo con las funcionalidades adicionadas.[43]

Los decoradores empleados en la implementación del sistema fueron `@login` y `@permission_required`, ambos son utilizados para la seguridad de la aplicación, `@login` consiste en obligar a que un usuario se encuentre autenticado para poder realizar operaciones en el sistema y `@permission_required` consiste en delimitar cuáles son los permisos que debe tener el usuario autenticado para realizar una determinada operación. En el siguiente fragmento de código se muestra el uso de ambos decoradores, los cuales aseguran que el usuario este autenticado y tenga permisos necesarios para habilitar categorías de usuario:

Figura 7. Código de las URLs principales del proyecto (@login)

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('iniciar-sesion/', LoginView.as_view(template_name='login.html', redirect_authenticated_user=True), name='iniciar_sesion'),
    # path('iniciar-sesion/', MyLoginView.as_view(), name='iniciar_sesion'),
    path('cerrar-sesion/', logout_then_login, name='cerrar_sesion'),
    path('', InicioView.as_view(), name='inicio'),
]
```

Figura 8. Código para habilitar una categoría de usuario

```
class HabilitarCategoriaUsuarioView(PermissionRequiredMixin, View):
    permission_required = ['administrador', 'especialista']

    def get(self, request, *args, **kwargs):
        categoria_usuario = CategoriaUsuario.objects.get(id=self.kwargs['pk'])
        categoria_usuario.activo = True
        categoria_usuario.save()
        messages.add_message(request, messages.SUCCESS, "Categoría habilitada con éxito.")
        return JsonResponse({})
```

2.7 Conclusiones Parciales

Con la elaboración de los artefactos que define la metodología XP, se garantizó una organización en el desarrollo del sistema y se logró simplificar el trabajo del implementador. Se especificó la arquitectura de software y los patrones de diseño, lo permitió el diseño de un sistema que podrá ser mantenido, auditado y actualizado con relativa facilidad, por parte de los desarrolladores en futuras versiones. Se confeccionaron además las tarjetas CRC, las cuales permitieron establecer el marco de relaciones para las clases de diseños. Estos artefactos ayudaran a una mejor comprensión de la manera en que se encuentra estructurado el sistema, lo cual facilita su análisis y, de ser necesario, su modificación.

Capítulo III. Implementación y prueba del sistema”

El presente capítulo describe la implementación y prueba del sistema, lo cual forma parte de la fase de construcción del mismo para su futura entrega y correcto funcionamiento. La metodología XP tiene este proceso después de la fase de diseño, cuyo objetivo es el de efectuar análisis y exámenes en todo momento del desarrollo. Se realizan pruebas unitarias a la aplicación para presenciar su correcto funcionamiento y pruebas de aceptación para no desviarse del producto final esperado por el cliente.

3.1 Tareas de ingeniería.

La metodología XP, define que es muy importante y necesario mostrar al cliente las especificaciones solicitadas, para esto se toman las historias de usuarios y se transforman en tareas de ingeniería para su fácil comprensión. Se muestran sus respectivos campos:

Número de la tarea: Los números deben ser consecutivos.

Número de HU: Número de la historia de usuario a la que pertenece la tarea.

Nombre Tarea: Nombre que identifica a la tarea.

Tipo de Tarea: Las tareas pueden ser de: Desarrollo, Mejora, Otra (Especificar).

Puntos Estimados: Tiempo estimado en días que se le asignará a su desarrollo.

Fecha Inicio: Fecha en que se inicia el desarrollo de la tarea.

Fecha Fin: Fecha en que finaliza el desarrollo de la tarea.

Programador Responsable: Nombre y apellidos del programador.

Descripción: Breve descripción de la tarea.

Con los campos anteriores se muestran las siguientes tareas de ingeniería:

Tabla 7. Tarea de ingeniería 1: Autenticación de Usuario

Tarea de ingeniería	
Número de la tarea: 1	Número de HU: 1
Nombre de la Tarea: Autenticación de Usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Fecha Inicio: 10/11/2021	Fecha Fin: 11/11/2021
Programador Responsable: Harold Pérez Trujillo	
Descripción: El usuario debe introducir sus datos y en caso de ser correctos, podrá acceder a las acciones que permite su rol.	

Tabla 8. Tarea de ingeniería 2: Registrar Usuario

Tarea de ingeniería	
Número de la tarea: 2	Número de HU: 2
Nombre de la Tarea: Registrar Usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 11/11/2021	Fecha Fin: 12/11/2021
Programador Responsable: Harold Pérez Trujillo	
Descripción: Cuando el usuario desea poseer permisos como usuario del sistema el mismo si no posee cuenta a través del botón registrarse, puede comenzar a llenar los campos solicitados y si los mismos están bien será registrado en el sistema.	

Tabla 9. Tarea de ingeniería 3: Modificar Categoría de Usuario

Tarea de ingeniería	
Número de la tarea: 3	Número de HU: 3
Nombre de la Tarea: Modificar Categoría de Usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Fecha Inicio: 12/11/2021	Fecha Fin: 12/11/2021
Programador Responsable: Harold Pérez Trujillo	
Descripción: El usuario con permisos para modificar la categoría de usuario deseada, desde la página accede a las opciones del listado de categorías de usuario, a la opción del botón editar la cual le permitirá actualizar esta categoría de usuario .	

Tabla 10. Tarea de ingeniería 4: Eliminar Accidente Laboral

Tarea de ingeniería	
Número de la tarea: 4	Número de HU: 9
Nombre de la Tarea: Eliminar Accidente Laboral	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3

Fecha Inicio: 12/11/2021	Fecha Fin: 13/11/2021
Programador Responsable: Harold Pérez Trujillo	
Descripción: El usuario con permisos para eliminar un accidente laboral determinado, desde la página accede a las opciones del listado de accidentes laborales, a la opción del botón eliminar la cual le permitirá eliminar dicho accidente laboral.	

Tabla 11. Tarea de ingeniería 5: Detalles de Persona

Tarea de ingeniería	
Número de la tarea: 5	Número de HU: 10
Nombre de la Tarea: Detalles de Persona	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Fecha Inicio: 13/11/2021	Fecha Fin: 13/11/2021
Programador Responsable: Harold Pérez Trujillo	
Descripción: El usuario con permisos para acceder a los detalles de una persona determinada, desde la página accede a las opciones del listado de persona, a la opción del botón detalles la cual le mostrará todos los datos de esta persona.	

Tabla 12. Tarea de ingeniería 6: Buscar Peligro

Tarea de ingeniería	
Número de la tarea: 6	Número de HU: 11
Nombre de la Tarea: Buscar Peligro	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 13/11/2021	Fecha Fin: 15/11/2021
Programador Responsable: Harold Pérez Trujillo	

Descripción: El usuario con permisos para acceder a buscar o filtrar un peligro determinado, desde la página encima del listado de peligro, donde se encuentra una lupa que se nombra filtrar, accede a ella y una vez ahí le saldrán los campos por los cuál filtrar, selecciona por cuál o cuales desea realizar el filtro y la página le mostrará debajo en el listado los resultados encontrados.

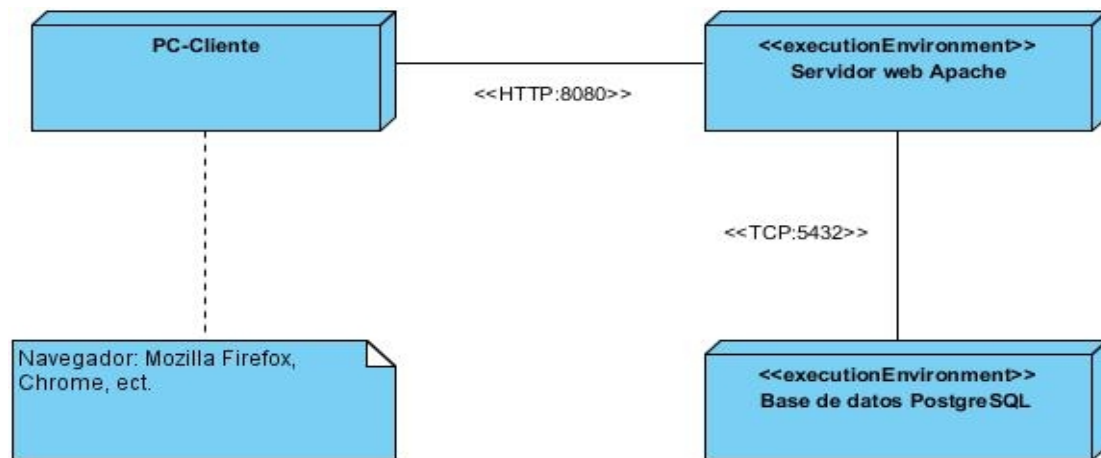
Tabla 13. Tarea de ingeniería 7: Eliminar Riesgos Seleccionados

Tarea de ingeniería	
Número de la tarea: 7	Número de HU: 12
Nombre de la Tarea: Eliminar Riesgos Seleccionados	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 15/11/2021	Fecha Fin: 16/11/2021
Programador Responsable: Harold Pérez Trujillo	
Descripción: El usuario con permisos para acceder a eliminar los o el riesgo seleccionado, desde la página en el listado de riesgo marca el o los riesgos que desee eliminar y luego accede al botón Eliminar seleccionados, que se encuentra arriba del listado luego de ejecutarlo selecciona aceptar y el o los riesgos seleccionados serán eliminados .	

3.1.1 Diagrama de despliegue

El diagrama de despliegue muestra las relaciones físicas entre los componentes: hardware y software, en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).[44]

Figura 9. Diagrama de despliegue



3.2 Estrategia de pruebas

Las pruebas de software son procesos que permiten verificar y revelar la calidad de un producto. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa.[14]

La creación de pruebas unitarias antes de que comience la codificación es un elemento clave del enfoque de XP. Las pruebas unitarias que se crean deben implementarse con el uso de una estructura que permita automatizarlas (de modo que puedan ejecutarse en repetidas veces y con facilidad).[14]

Las pruebas de aceptación XP, llamadas también como pruebas del cliente, son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptación se derivan de las historias de los usuarios que se han implementado.

3.2.1 Pruebas unitarias

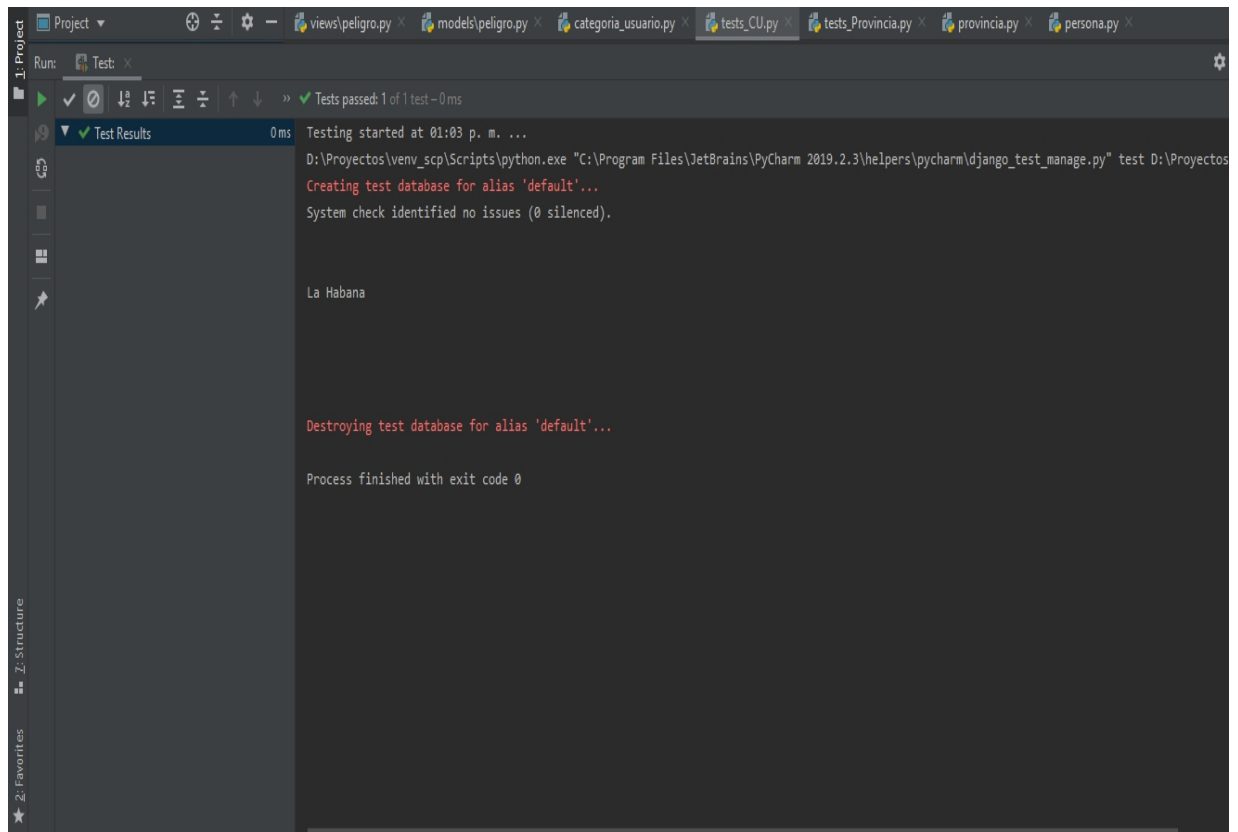
Las pruebas unitarias son realizadas a pequeñas porciones de código, por separados, para verificar su correcta funcionalidad, las mismas se pueden ir efectuando desde el comienzo de la implementación, no necesariamente se tiene que esperar al finalizar el software.

La prueba de unidad enfoca los esfuerzos de verificación en la unidad más pequeña del diseño de software: el componente o módulo de software. Al usar la descripción del diseño de componente como guía, las rutas de control importantes se prueban para descubrir errores dentro de la frontera del módulo. La relativa complejidad de las pruebas y los errores que descubren están limitados por el ámbito restringido que se establece para la prueba de unidad. Las pruebas de unidad se enfocan en la lógica de procesamiento interno y de las estructuras de datos dentro de las fronteras de un componente. Este tipo de pruebas puede realizarse en paralelo para múltiples componentes.[34]

Las pruebas unitarias de Django utilizan una biblioteca de Python: "unittest". Para la implementación de las pruebas realizadas a la plataforma web, se utilizó la clase

base de prueba TestCase y posteriormente se escribieron métodos separados para verificar que la funcionalidad específica funcione correctamente. A continuación, se ejemplifican las pruebas realizadas a las clases Categoría usuario y Provincia mediante las clases **CategoriaUsuarioTestCase** y **ProvinciaTestCase** (Ver Anexo III Pruebas Unitarias), donde se obtienen resultados satisfactorios:

Figura 10. Test realizado a la clase ProvinciaTestCase



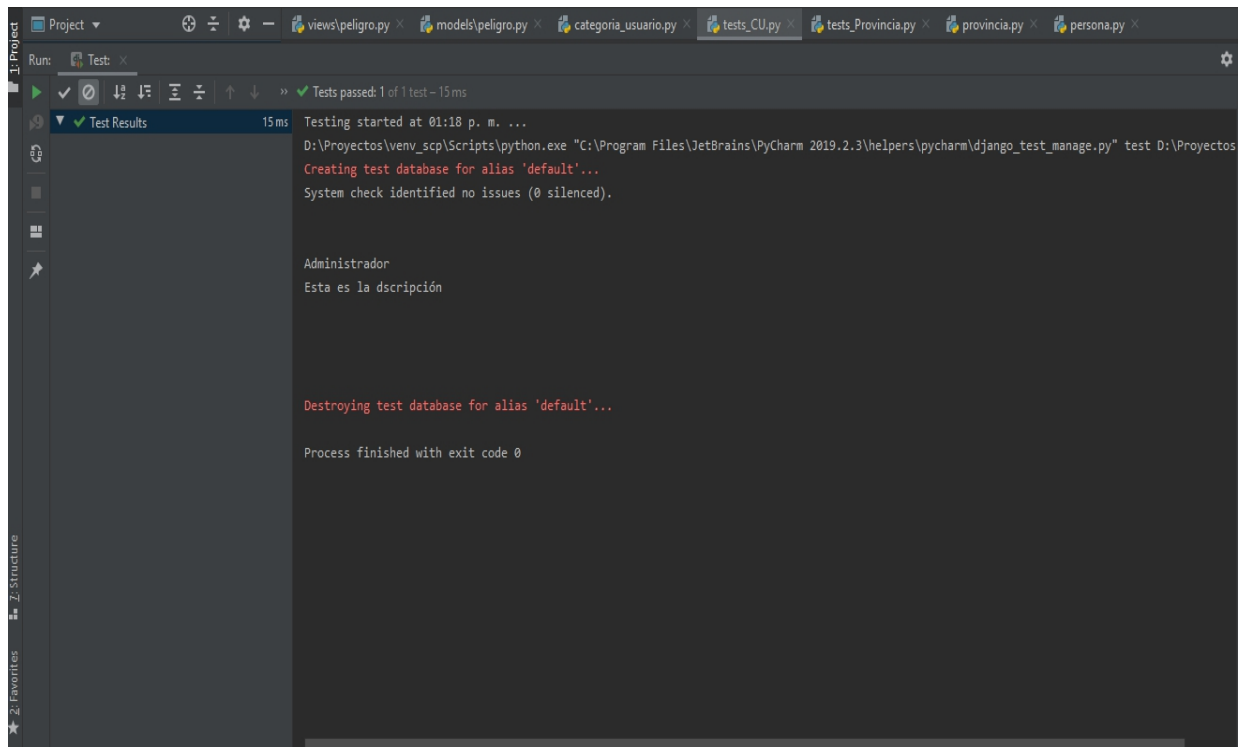
```
Run: Test: x
Tests passed: 1 of 1 test - 0 ms
Test Results 0 ms
Testing started at 01:03 p. m. ...
D:\Proyectos\venv_scp\Scripts\python.exe "C:\Program Files\JetBrains\PyCharm 2019.2.3\helpers\pycharm\django_test_manage.py" test D:\Proyectos
Creating test database for alias 'default'...
System check identified no issues (0 silenced).

La Habana

Destroying test database for alias 'default'...

Process finished with exit code 0
```

Figura 11. Test realizado a la clase CategoríaUsuarioTestCase



3.2.2 Pruebas de aceptación

Las pruebas de aceptación son creadas en base de las Historias de Usuarios, en cada ciclo de la iteración del desarrollo. Las pruebas de aceptación XP, también llamadas pruebas del cliente, son especificadas por el mismo cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del mismo. Por lo tanto, son los clientes los responsables de verificar que los datos de estas pruebas sean correctos. Se muestran a continuación sus campos:

Código: Representa el caso de prueba, incluye el número de la historia de usuario y de la prueba.

Historia de usuario: Número de la historia de usuario.

Nombre: Incluye el nombre de la historia de usuario.

Descripción: Acción que debe realizar el sistema.

Condiciones de ejecución: Características y elementos que debe contener el sistema para ejecutar el caso de prueba.

Entrada/Pasos de ejecución: Incluye los pasos necesarios para realizar la prueba.

Resultado esperado: Respuesta que el sistema debe dar ante la ejecución de la prueba.

Evaluación de prueba: Clasificación de la prueba en satisfactoria o no satisfactoria.

Tabla 14. Prueba de Aceptación: Autenticar usuario

Caso de prueba de Aceptación	
Código:HU1_CP1	Historia de usuario:1
Nombre: Autenticar usuario.	
Descripción: Prueba para la funcionalidad autenticar usuario.	
Condiciones de ejecución: El usuario deberá introducir sus credenciales para acceder al sistema.	
Entrada/Pasos de ejecución: Introducir nombre de usuario y contraseña y luego pulsar el botón acceder .	
Resultado esperado: Si el usuario se encuentra registrado en el sistema accederá, en caso contrario no obtendrá acceso al sistema.	
Evaluación de prueba: Satisfactoria.	

Tabla 15. Prueba de Aceptación: Gestionar usuario

Caso de prueba de Aceptación	
Código:HU2_CP2	Historia de usuario:2
Nombre: Gestionar usuario.	
Descripción: Prueba para la funcionalidad gestionar usuario. Un usuario con permisos para gestionar usuario, podrá acceder a dicha gestión desde la barra de menú, entrando a la pestaña de usuarios, ahí accede a gestionar toda la información de los usuarios sobre los que tenga otorgado permisos.	
Condiciones de ejecución: Poseer permisos para gestionar usuario.	
Entrada/Pasos de ejecución: Identificar que usuario desea gestionar, ahí le aparecerán todas las opciones para gestionarlo.	
Resultado esperado: Una vez realizada cualquier opción de gestión, se espera un mensaje de confirmación de la operación realizada.	
Evaluación de prueba: Satisfactoria.	

Tabla 16. Prueba de Aceptación: Gestionar accidente laboral

Caso de prueba de Aceptación	
Código:HU9_CP2	Historia de usuario:9
Nombre: Gestionar accidente laboral.	

Descripción: Prueba para la funcionalidad gestionar accidente laboral. Un usuario con permisos para gestionar accidente laboral, podrá acceder a dicha gestión desde la barra de menú, entrando a la pestaña de accidentes laborales y dentro a la pestaña accidentes laborales, ahí accede a gestionar toda la información de los accidentes laborales sobre los que tenga otorgado permisos.
Condiciones de ejecución: Poseer permisos para gestionar accidente laboral.
Entrada/Pasos de ejecución: Identificar que accidente laboral desea gestionar, ahí le aparecerán todas las opciones para gestionarlo.
Resultado esperado: Una vez realizada cualquier opción de gestión, se espera un mensaje de confirmación de la operación realizada.
Evaluación de prueba: Satisfactoria.

Tabla 17. Prueba de Aceptación: Gestionar persona

Caso de prueba de Aceptación	
Código:HU10_CP3	Historia de usuario:10
Nombre: Gestionar persona.	
Descripción: Prueba para la funcionalidad gestionar persona. Un usuario con permisos para gestionar persona, podrá acceder a dicha gestión desde la barra de menú, entrando a la pestaña de accidentes laborales y dentro a la pestaña personas, ahí accede a gestionar toda la información de las personas sobre las que tenga otorgado permisos.	
Condiciones de ejecución: Poseer permisos para gestionar persona.	
Entrada/Pasos de ejecución: Identificar que persona desea gestionar, ahí le aparecerán todas las opciones para gestionarla.	
Resultado esperado: Una vez realizada cualquier opción de gestión, se espera un mensaje de confirmación de la operación realizada.	
Evaluación de prueba: Satisfactoria.	

Tabla 18. Prueba de Aceptación: Gestionar peligro

Caso de prueba de Aceptación	
Código:HU11_CP4	Historia de usuario:11
Nombre: Gestionar peligro.	
Descripción: Prueba para la funcionalidad gestionar peligro. Un usuario con permisos para gestionar peligro, podrá acceder a dicha gestión desde la barra de menú, entrando a la pestaña de peligros y riesgos y dentro a la pestaña peligros, ahí accede a gestionar toda la información de los peligros sobre las que tenga	

otorgado permisos.
Condiciones de ejecución: Poseer permisos para gestionar peligro.
Entrada/Pasos de ejecución: Identificar que peligro desea gestionar, ahí le aparecerán todas las opciones para gestionarlo.
Resultado esperado: Una vez realizada cualquier opción de gestión, se espera un mensaje de confirmación de la operación realizada.
Evaluación de prueba: Satisfactoria.

Tabla 19. Prueba de Aceptación: Gestionar riesgo

Caso de prueba de Aceptación	
Código:HU12_CP5	Historia de usuario:12
Nombre: Gestionar riesgo.	
Descripción: Prueba para la funcionalidad gestionar riesgo. Un usuario con permisos para gestionar riesgo, podrá acceder a dicha gestión desde la barra de menú, entrando a la pestaña de peligros y riesgos y dentro a la pestaña riesgos, ahí accede a gestionar toda la información de los riesgos sobre las que tenga otorgado permisos.	
Condiciones de ejecución: Poseer permisos para gestionar riesgo.	
Entrada/Pasos de ejecución: Identificar que riesgo desea gestionar, ahí le aparecerán todas las opciones para gestionarlo.	
Resultado esperado: Una vez realizada cualquier opción de gestión, se espera un mensaje de confirmación de la operación realizada.	
Evaluación de prueba: Satisfactoria.	

3.2.3 Resultado de las Pruebas de Aceptación

En total se realizaron un total de 6 casos de pruebas de aceptación. Estas fueron realizadas pasando por cada iteración, supervisando de esta manera el cumplimiento de cada tarea de ingeniería correspondiente a estas. **Anexo IV Resultado de las Pruebas de Aceptación**, se muestra una gráfica con el grado de aceptación obtenido al realizar las pruebas.

Al culminar el proceso de pruebas por iteración, se detectaron un grupo de no conformidades que se fueron solucionando antes de comenzar la siguiente etapa, permitiendo que al culminar la última iteración de pruebas la plataforma web tuviera un gran por ciento de aceptación ante el cliente. En la primera iteración se detectaron 2 no conformidades relacionadas con el diseño de la interfaz de usuario y

con la validación de los datos de registro. Para la segunda iteración se detectaron un total de 2 no conformidades relacionadas con el modelo de Riesgo y diseño de la interfaz de usuario. En la tercera iteración hubo un total de 2 no conformidades relacionadas con los filtros de búsqueda.

Al término de cada una de las iteraciones se realizaron pruebas de regresión, con el objetivo de asegurar que, al corregir las no conformidades detectadas, estas no introdujeran nuevos errores en la solución.

3.3 Estándares de codificación

Los estándares de codificación son buenas prácticas de programación que deben tenerse en cuenta a la hora de desarrollar un software, así brindando más facilidad al programador y a su vez a futuras actualizaciones del software, disminuye el número de decisiones que debe tomar el programador, ya que están definidas las posibles acciones que puede realizar.

A continuación, los estándares utilizados en el proyecto:

1. Comentar código: Consiste en agregar comentarios que explique en que consiste un determinado fragmento de código, con el fin de ayudar y agilizar las futuras versiones de la aplicación.
2. Formato y legibilidad: Se establece la posición en que irán las llaves dentro del código y la cantidad de espacios o tabulaciones.
3. Usar nombres de variables y rutas que aporten al entendimiento del código.

3.4 Conclusiones Parciales

En el presente capítulo se realizaron las pruebas a la aplicación Sistema de Gestión para el Control de la Seguridad y Salud en el Trabajo tomando en cuenta las fases implementación y pruebas de la metodología XP. Con del desarrollo de las funcionalidades del sistema fueron realizadas las pruebas correspondientes. Las no conformidades encontradas fueron solucionadas en las mismas iteraciones, por ello todas las tareas de ingeniería fueron reflejadas para la mejor comprensión del cliente. De esta manera se obtuvo una aplicación web con la calidad y las funcionalidades solicitadas por el cliente.

Conclusiones Generales

Una vez desarrollada la presente investigación, cumplidas las tareas de investigación y el objetivo general, se arriba a las siguientes conclusiones:

- ◆ El desarrollo del marco teórico permitió revisar los antecedentes relacionados al tema de la investigación relacionado con la gestión de la Seguridad y Salud en el Trabajo e identificar los conceptos asociados de la investigación.
- ◆ Con el análisis de las aplicaciones web NOVASOFT-GTH, SARA- Sistema de Seguridad y Salud en el Trabajo y la apk GESTIÓN SALUD Y SEGURIDAD EN EL TRABAJO que gestionan la Seguridad y Salud en el Trabajo, ninguno de ellos reúne los requisitos necesarios para ser utilizados directamente como propuesta de solución, por lo que debe desarrollarse una nueva, pero se obtuvieron las características para tenerse en cuenta en el sistema desarrollado.
- ◆ A partir de análisis se identificó que la metodología a desarrollar es XP, las tecnologías y lenguajes son (HTML, CSS, JavaScript, JQuery, Bootstrap, Python, Django), como sistema gestor PostgreSQL y herramientas (Visual Paradigm, PyCharm).
- ◆ Se profundizó en el proceso de desarrollo bajo el patrón de diseño Model-Template-View. Con el empleo del patrón de diseño Model-Template-View y los patrones de diseño Grasp y Gof, permitieron el correcto diseño de la aplicación.
- ◆ Se validó el sistema haciendo uso de las pruebas unitarias y de aceptación comprobando funcionalidades del sistema y directamente al código, para realizar las pruebas unitarias con Django se utilizó una biblioteca de Python: "unittest". La implementación de la solución propuesta, luego de ser validada y verificada a partir de las pruebas definidas, permitió comprobar la conformidad con los requisitos especificados y además, satisfacer con las necesidades del cliente.

Recomendaciones

Se recomienda para posibles investigaciones:

Extender el sistema en próximas versiones al sector no estatal.

Referencias bibliográficas

- [1] LIZÁRRAGA-PORTUGAL, Carlos-Augusto. Seguridad y Salud en el Trabajo. 2018.
- [2] SALUD, O. I. T. seguridad en trabajo en América Latina y el Caribe. 2014.
- [3] SOCARRÁS, Gustavo Manuel Céspedes; CUMBRERA, Jorge Manuel Martínez. Un análisis de la seguridad y salud en el trabajo en el sistema empresarial cubano. *Revista latinoamericana de derecho social*, 2016, vol. 22, p. 1-46.
- [4] CHALLENGER, Antony. La aplicación del concepto del sistema socio-ecológico: alcances, posibilidades y limitaciones en la gestión ambiental de México. *Investigación ambiental Ciencia y política pública*, 2015, vol. 6, no2.
- [5] PLASENCIA ASOREY, Carolina. El Sistema de Control Interno: garantía del logro de los objetivos. *Medisan*, 2010, vol. 14, no 5, p. 0-0.
- [6] Gaceta oficial No.29, artículo 126, capítulo XI del Código del trabajo.
- [7] Gaceta oficial No.29, artículo 128, capítulo XI del Código del trabajo.
- [8] Gaceta oficial No.29, artículo 129, capítulo XI del Código del trabajo.
- [9] Gaceta oficial No.29, artículo 130, capítulo XI del Código del trabajo.
- [9] PRESSMAN, R.S. y CONTRERAS, T.B., 2010. Desarrollo ágil. *Pablo Roig Vásquez. Ingeniería del software. Ciudad de México: McGraw-Hill.*
- [10] GESTIÓN SALUD Y SEGURIDAD EN EL TRABAJO FOR ANDROID. [En línea]. Disponible en: <https://m.apkpure.com/es/gesti%C3%B3n-salud-y-seguridad-en-el-trabajo/aplus.gestinsaludyseguridad>. [Accedido: 15-jul-2021].
- [11] NOVASOFT-GTH. [En línea]. Disponible en: <https://quiatic.com/co/50-gestion-de-recursos-humanos-y-nomina/1211-novasoft-gth-software-de-nomina-gestion-humana-sg-sst-y-contratistas-plataforma-para-la-administracion-de-personas>. [Accedido: 15-jul-2021].
- [12] SARA - Sistema de Seguridad y Salud en el Trabajo. [En línea]. Disponible en: <https://quiatic.com/co/19-software-para-salud-ocupacional/277-sara-sistema-de-seguridad-y-salud-en-el-traba>. [Accedido: 15-jul-2021].
- [13] PRESSMAN, R.S. y CONTRERAS, T.B., 2010. Desarrollo ágil. *Pablo Roig Vásquez. Ingeniería del software. Ciudad de México: McGraw-Hill.*
- [14] PRESSMAN, R.S. y CONTRERAS, T.B., 2010. Desarrollo ágil. *Pablo Roig Vásquez. Ingeniería del software. Ciudad de México: McGraw-Hill*, pp. 61, 62 ,69, 139, 190.
- [15] RODRÍGUEZ, Y.R., KILE, S.B.F. y ROMERO, A.E.R., 2013. “Aplicación de la estrella de Boehm y Turner al proyecto laboratorios virtuales. UCV-HACER. *Revista de Investigación y Cultura*”, vol. 2, no. 1, pp. 20–29.

- [16] PÉREZ, Javier Eguíluz. Introducción a XHTML. 2019.
- [17] GUAPI AUQUILLA, María José. Diseño metodológico para el desarrollo de interfaces gráficas en páginas web utilizando los lenguajes HTML 5 y CSS 3. 2018. Tesis de Licenciatura. Riobamba.
- [18] PÉREZ, Javier Eguíluz. introduccion a JavaScript. 2019.
- [19] PÉREZ, Javier Eguíluz. Introducción a CSS. Librosweb. es (editorial digital), 2009.
- [20] RAMÍREZ, Cristal Esmeralda Germain; ROBLES, Rodolfo Ostos; CELIS, Brandon Aguilar. Desarrollo de aplicaciones web utilizando JavaScript. 2020.
- [21] Forcier, Jeff, Bissex, Paul, & Chun, Wesley. (2008). Weley. "Python web developmetn with Django. Addison-Wesley Professional".
- [22] SISALIMA TAPIA, Hilda María; TENELEMA ARIAS, Esthela Nataly. Análisis de rendimiento y productividad Cross-Browser HTLM. dom de los frameworks Mootools, Jquery y Yui. Caso práctico: Evaluación de destrezas en el área de inglés UESTAR. 2014. Tesis de Licenciatura.
- [23] JÁCOME AYALA, Paulina Johanna. Benchmarking de los Frameworks Opensource: Bootstrap y Uikit Sistema administrador y de registro para el uso de laboratorios de computación. 2016. Tesis de Licenciatura.
- [24] Montero, S. (2016). Bootstrap 4 By Example. Packt Publishing Ltd.
- [25] CUMBA ARMIJOS, Paúl David; BARRENO PILCO, Byron Augusto. Análisis de PYTHON con Django frente a Ruby on Rails para desarrollo ágil de aplicaciones web. Caso práctico: DECH. 2013. Tesis de Licenciatura.
- [26] FERNÁNDEZ, Arturo. Python 3 al descubierto. Alfaomega grupo Editor, 2013.
- [27] TUBAY AVEIGA, Bryan Steveen. DISEÑO DE UNA PÁGINA WEB INFORMATIVA PARA LA PUBLICIDAD DE LA EMPRESA MAKTUB DE LA CIUDAD DE GUAYAQUIL 2019. 2020. Tesis de Licenciatura. Instituto Superior Tecnológico Bolivariano de Tecnología.
- [28] TENE LA ROSA, Dina Noemi; MONTESE ALONZO, Fanny Magdalena. Diseño de un prototipo Web interactivo basado en técnicas de Machine Learning y razonamiento probabilístico para pronosticar mediante los factores de riesgo la enfermedad de la diabetes. 2019. Tesis de Licenciatura. Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Sistemas Computacionales.
- [29] RAMÍREZ, RAQUEL ZAMBRANO; MEDIO, CICLO FORMATIVO GRADO. Sistemas Gestores de Bases de Datos. *Innovación y Experiencias Educativas*, 2008.
- [30] LÓPEZ HERRERA, Patricia. Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL. 2016.
- [31] CORREA REAL, Luis Germán. Análisis comparativo entre las base datos no relacional MongoDB con la base de datos Postgresql, sistema para la gestión de clientes y registros de pagos de la Clínica Odontológica Ortho Dent. 2015. Tesis de Licenciatura.

[32] CHAVARRÍA-BÁEZ, Lorena; ROJAS, Nancy OCOTITLA. Sobre el uso de herramientas CASE para la enseñanza de bases de datos. *Sistemas, Cibernetica e Informática*, 2016, vol. 13, no 2, p. 51-56.

[33] Gutiérrez, E. (2019). Módulos de lógica proposicional y circuitos lógicos para la plataforma web interactiva y experimental "Sophia". Universidad de las Ciencias Informáticas, La Habana.

[34] Pressman, Roger S. *Ingeniería de Software. Un enfoque práctico*. [ed.] María Teresa Zapata Terrazas. Séptima. México : s.n., 2010. p. 736. ISBN: 978-607-15-0314-5.

[35] ALARCÓN VALERO, Faustino; ALEMANY DÍAZ, María del Mar; JEANPIERRE, FREDERIC. Metodología para la reingeniería basada en el modelado de procesos de negocio. *DYNA Ingeniería e Industria*, 2012, vol. 87, no 5, p. 566-573.

[36] Luis Calabria, Pablo Piriz. 2003. Metodología XP. Uruguay: s.n., 2003.

[37] MENZINSKY, Alexander, et al. Historias de Usuario. Ingeniería de requisitos ágil, 2018.

[38] *Phython - Django Framework de desarrollo web para perfeccionistas basado en el modelo MTV*. Condori Ayala, José Luis. La Paz : s.n., Noviembre 2012, Revista de Información, Tecnología y Sociedad. ISSN: 1997-4044.

[39] Holovaty, Adrian and K. Moss, Jacob. *The Definitive Guide to Django: Web Development Done Right*. Segunda. New York : s.n. p. 361. ISBN: 978-1-4302-19361.

[40] Infante Montero, Sergio. Desarrollo web. Curso Django: Entendiendo como trabaja Django. [Online] Abril 30, 2012. [Cited: Febrero 24, 2017.] <http://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django/>.

[41] Holovaty, Adrian and Kaplan Moss, Jacob. Libros Web. El libro de Django. *El patrón de diseño MTV*. [Online] 2016-2017. [Cited: Febrero 19, 2017.]

https://librosweb.es/libro/django_1_0/capitulo_5/el_patron_de_diseno_mtv.html.

[42] Sommerville, Ian. *Sommerville. Ingeniería de software*. [ed.] Luis M. Cruz Castillo. Novena. México : Pearson Educación de México, S.A. de C.V., 2011. p. 792. ISBN: 978-607-32-0603-7.

[43] TAMAYO, E. S. G. Y. B. L. Sistema para la Calificación Automática en Competencias de Programación. La Habana, Cuba : s.n., Junio 2013. Facultad 1. Universidad de las Ciencias Informáticas (UCI).

[44] I. Sommerville, «Software engineering 9th Edition», *ISBN-10*, vol. 137035152, p.18, 2011.

Bibliografía

MAIDA, Esteban Gabriel; PACIENZA, Julián. Metodologías de desarrollo de software. 2015.

CORA, Hidelvys Cantero, et al. Procedimiento para el diagnóstico de seguridad y salud del trabajo (SST). Observatorio de la Economía Latinoamericana, 2012, no 167.

Penadés, Patricio Letelier. 2010. "Metodologías ágiles para el desarrollo de software: Extrema Programming". Valencia: s.n., 2010

LIZÁRRAGA-PORTUGAL, Carlos-Augusto. Seguridad y Salud en el Trabajo. 2018.

SALUD, O. I. T. seguridad en trabajo en América Latina y el Caribe. 2014.

SOCARRÁS, Gustavo Manuel Céspedes; CUMBREIRA, Jorge Manuel Martínez. Un análisis de la seguridad y salud en el trabajo en el sistema empresarial cubano. *Revista latinoamericana de derecho social*, 2016, vol. 22, p. 1-46.

CHALLENGER, Antony. La aplicación del concepto del sistema socio-ecológico: alcances, posibilidades y limitaciones en la gestión ambiental de México. *Investigación ambiental Ciencia y política pública*, 2015, vol. 6, no2.

PLASENCIA ASOREY, Carolina. El Sistema de Control Interno: garantía del logro de los objetivos. *Medisan*, 2010, vol. 14, no 5, p. 0-0.

PRESSMAN, R.S. y CONTRERAS, T.B., 2010. Desarrollo ágil. *Pablo Roig Vásquez. Ingeniería del software. Ciudad de México: McGraw-Hill*, pp. 61,62.

FERNÁNDEZ, Arturo. Python 3 al descubierto. Alfaomega grupo Editor, 2013.

RODRÍGUEZ, Y.R., KILE, S.B.F. y ROMERO, A.E.R., 2013. "Aplicación de la estrella de Boehm y Turner al proyecto laboratorios virtuales. UCV-HACER. *Revista de Investigación y Cultura*", vol. 2, no. 1, pp. 20–29.

Javier Eguiluz Pérez, "Introducción a JavaScript", Versión Impresa 2009.

Forcier, Jeff, Bissex, Paul, & Chun, Wesley. (2008). Weley. "Python web developmetn with Django. Addison-Wesley Professional".

TENE LA ROSA, Dina Noemi; MONTESE ALONZO, Fanny Magdalena. Diseño de un prototipo Web interactivo basado en técnicas de Machine Learning y razonamiento probabilístico para pronosticar mediante los factores de riesgo la enfermedad de la diabetes. 2019. Tesis de Licenciatura. Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Sistemas Computacionales.

RAMÍREZ, Cristal Esmeralda Germain; ROBLES, Rodolfo Ostos; CELIS, Brandon Aguilar. Desarrollo de aplicaciones web utilizando JavaScript. 2020.

TUBAY AVEIGA, Bryan Steeven. DISEÑO DE UNA PÁGINA WEB INFORMATIVA PARA LA PUBLICIDAD DE LA EMPRESA MAKTUB DE LA CIUDAD DE GUAYAQUIL 2019. 2020. Tesis de Licenciatura. Instituto Superior Tecnológico Bolivariano de Tecnología.

JÁCOME AYALA, Paulina Johanna. Benchmarking de los Frameworks Opensource: Bootstrap y Uikit Sistema administrador y de registro para el uso de laboratorios de computación. 2016. Tesis de Licenciatura.

Montero, S. (2016). Bootstrap 4 By Example. Packt Publishing Ltd.

Gutiérrez, E. (2019). Módulos de lógica proposicional y circuitos lógicos para la plataforma web interactiva y experimental "Sophia". Universidad de las Ciencias Informáticas, La Habana.

MÁRQUEZ, Glenda Rivas. Modelos contemporáneos de control interno. Fundamentos teóricos. Observatorio Laboral Revista Venezolana, 2011, vol. 4, no 8, p. 115-136.

PADRÓN, Raúl, et al. Control Social del Estado. Corporación Universitaria Rafael Núñez, 2020.

SISALIMA TAPIA, Hilda María; TENELEMA ARIAS, Esthela Nataly. Análisis de rendimiento y productividad Cross-Browser HTLM. dom de los frameworks Mootools, Jquery y Yui. Caso práctico: Evaluación de destrezas en el área de inglés UESTAR. 2014. Tesis de Licenciatura.

FRANCO GARCÍA, Leidy Camila, et al. Propuesta de mejoramiento para el sistema de gestión de seguridad y salud en el trabajo basado en la norma ISO 45001: 2018 para la empresa de Outsourcing SA. 2020.

IDROVO, Alvaro Javier; ALVAREZ-CASADO, Enrique. Asumiendo el concepto de salud. Salud UIS, 2014, vol. 46, no 3.

TUMBACO, Sandra Leonor Chiquito; ALCIVAR, Byron José Loo; MERCHÁN, Sonia Melissa Rodríguez. Sistema de Seguridad y Salud en el Trabajo. Transición de las OHSAS 18001: 2007 a la nueva ISO 45001. Revista Publicando, 2016, vol. 3, no 9, p. 638-648.

CUMBA ARMIJOS, Paúl David; BARRENO PILCO, Byron Augusto. Análisis de PYTHON con Django frente a Ruby on Rails para desarrollo ágil de aplicaciones web. Caso práctico: DECH. 2013. Tesis de Licenciatura.

GUAPI AUQUILLA, María José. Diseño metodológico para el desarrollo de interfaces gráficas en páginas web utilizando los lenguajes HTML 5 y CSS 3. 2018. Tesis de Licenciatura. Riobamba.

HUAMÁN RONDINEL, Milagros Esperanza. Implementación del sistema de gestión de seguridad y salud en el trabajo en base a la norma ISO 45001: 2018 en la empresa Constructora Alvaro Ingenieros EIRL, Piura-2019. 2020.

RAMÍREZ, RAQUEL ZAMBRANO; MEDIO, CICLO FORMATIVO GRADO. Sistemas Gestores de Bases de Datos. *Innovación y Experiencias Educativas*, 2008.

LÓPEZ HERRERA, Patricia. Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL. 2016.

CHAVARRÍA-BÁEZ, Lorena; ROJAS, Nancy OCOTITLA. Sobre el uso de herramientas CASE para la enseñanza de bases de datos. *Sistemas, Cibernetica e Informática*, 2016, vol. 13, no 2, p. 51-56.

MENZINSKY, Alexander, et al. Historias de Usuario. Ingeniería de requisitos ágil, 2018.

Pressman, Roger S. *Ingeniería de Software. Un enfoque práctico*. [ed.] María Teresa Zapata Terrazas. Séptima. México : s.n., 2010. p. 736. ISBN: 978-607-15-0314-5.

CORREA REAL, Luis Germán. Análisis comparativo entre las base datos no relacional MongoDB con la base de datos PostgreSQL, sistema para la gestión de clientes y registros de pagos de la Clínica Odontológica Ortho Dent. 2015. Tesis de Licenciatura.

Gaceta Oficial No.29 Extraordinaria de 17 de junio de 2014, Código del Trabajo de la República de Cuba.

Luis Calabria, Pablo Piriz. 2003. Metodología XP. Uruguay: s.n., 2003.

ALARCÓN VALERO, Faustino; ALEMANY DÍAZ, María del Mar; JEANPIERRE, FREDERIC. Metodología para la reingeniería basada en el modelado de procesos de negocio. DYNA Ingeniería e Industria, 2012, vol. 87, no 5, p. 566-573.

GESTIÓN SALUD Y SEGURIDAD EN EL TRABAJO FOR ANDROID. [En línea]. Disponible en: <https://m.apkpure.com/es/gesti%C3%B3n-salud-y-seguridad-en-el-trabajo/aplus.gestinsaludyseguridad>. [Accedido: 15-jul-2021].

NOVASOFT-GTH. [En línea]. Disponible en: <https://guiatic.com/co/50-gestion-de-recursos-humanos-y-nomina/1211-novasoft-gth-software-de-nomina-gestion-humana-sg-sst-y-contratistas-plataforma-para-la-administracion-de-personas>. [Accedido: 15-jul-2021].

SARA - Sistema de Seguridad y Salud en en Trabajo. [En línea]. Disponible en: <https://guiatic.com/co/19-software-para-salud-ocupacional/277-sara-sistema-de-seguridad-y-salud-en-el-traba>. [Accedido: 15-jul-2021].

Python - Django Framework de desarrollo web para perfeccionistas basado en el modelo MTV. Condori Ayala, José Luis. La Paz : s.n., Noviembre 2012, Revista de Información, Tecnología y Sociedad. ISSN: 1997-4044.

Holovaty, Adrian and K. Moss, Jacob. *The Definitive Guide to Django: Web*

Development Done Right. Segunda. New York : s.n. p. 361. ISBN: 978-1-4302-19361.

Infante Montero, Sergio. Desarrollo web. Curso Django: Entendiendo como trabaja Django. [Online] Abril 30, 2012. [Cited: Febrero 24, 2017.] <http://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django/>.

Holovaty, Adrian and Kaplan Moss, Jacob. Libros Web. El libro de Django. *El patrón de diseño MTV*. [Online] 2016-2017. [Cited: Febrero 19, 2017.]

https://librosweb.es/libro/django_1_0/capitulo_5/el_patron_de_diseno_mtv.html.

Sommerville, Ian. *Sommerville. Ingeniería de software*. [ed.] Luis M. Cruz Castillo. Novena. México : Pearson Educación de México, S.A. de C.V., 2011. p. 792. ISBN: 978-607-32-0603-7.

TAMAYO, E. S. G. Y. B. L. Sistema para la Calificación Automática en Competencias de Programación. La Habana, Cuba : s.n., Junio 2013. Facultad 1. Universidad de las Ciencias Informáticas (UCI).

Anexos

1. Historias de usuario

Historia de usuario	
Número: HU_2	Nombre historia de usuario: Gestionar usuario
Modificación de historia de usuario: 2	
Programador: Harold Pérez Trujillo	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 9 días
Riesgo en Desarrollo: Alta	Tiempo Real: 8
Descripción: muestra los campos correspondientes a registrar, iniciar sesión, cerrar sesión, listar, modificar, detalles, eliminar, eliminar seleccionados, habilitar, deshabilitar, cambiar contraseña.	
Observaciones: el campo contenido está seleccionado por defecto.	

Historia de usuario	
Número: HU_3	Nombre historia de usuario: Gestionar categoría de usuario
Modificación de historia de usuario: 2	
Programador: Harold Pérez Trujillo	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 10 días
Riesgo en Desarrollo: Alta	Tiempo Real: 9
Descripción: muestra los campos correspondientes a registrar, listar, modificar, eliminar, eliminar seleccionadas, habilitar, deshabilitar.	
Observaciones: el campo contenido está seleccionado por defecto.	

Historia de usuario	
Número: HU_4	Nombre historia de usuario: Gestionar provincia
Modificación de historia de usuario: 1	
Programador: Harold Pérez Trujillo	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 10 días
Riesgo en Desarrollo: Alta	Tiempo Real: 9
Descripción: muestra los campos correspondientes a registrar, listar, modificar, eliminar, eliminar seleccionadas, habilitar, deshabilitar.	
Observaciones: el campo contenido está seleccionado por defecto.	

Historia de usuario	
Número: HU_5	Nombre historia de usuario: Gestionar municipio
Modificación de historia de usuario: 1	
Programador: Harold Pérez Trujillo	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 10 días
Riesgo en Desarrollo: Alta	Tiempo Real: 8
Descripción: muestra los campos correspondientes a registrar, listar, modificar, eliminar, buscar, eliminar seleccionados, habilitar, deshabilitar.	
Observaciones: el campo contenido está seleccionado por defecto.	

Historia de usuario	
Número: HU_6	Nombre historia de usuario: Gestionar entidad

Modificación de historia de usuario: 3	
Programador: Harold Pérez Trujillo	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 10 días
Riesgo en Desarrollo: Alta	Tiempo Real: 10
Descripción: muestra los campos correspondientes a registrar, listar, modificar, eliminar, buscar, eliminar seleccionadas, habilitar, deshabilitar.	
Observaciones: el campo contenido está seleccionado por defecto.	

Historia de usuario	
Número: HU_7	Nombre historia de usuario: Gestionar sector
Modificación de historia de usuario: 2	
Programador: Harold Pérez Trujillo	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 9 días
Riesgo en Desarrollo: Alta	Tiempo Real: 9
Descripción: muestra los campos correspondientes a registrar, listar, modificar, eliminar, buscar, eliminar seleccionadas, habilitar, deshabilitar.	
Observaciones: el campo contenido está seleccionado por defecto.	

Historia de usuario	
Número: HU_8	Nombre historia de usuario: Gestionar organismo
Modificación de historia de usuario: 2	
Programador: Harold Pérez Trujillo	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 9 días
Riesgo en Desarrollo: Alta	Tiempo Real: 8

Descripción: muestra los campos correspondientes a registrar, listar, modificar, eliminar, buscar, eliminar seleccionados, habilitar, deshabilitar.

Observaciones: el campo contenido está seleccionado por defecto.

Historia de usuario

Número: HU_9	Nombre historia de usuario: Gestionar accidente laboral
---------------------	--

Modificación de historia de usuario: 3

Programador: Harold Pérez Trujillo	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 10 días
Riesgo en Desarrollo: Alta	Tiempo Real: 10

Descripción: muestra los campos correspondientes a registrar, listar, modificar, eliminar, detalles, buscar, eliminar seleccionados, habilitar, deshabilitar.

Observaciones: el campo contenido está seleccionado por defecto.

Historia de usuario

Número: HU_10	Nombre historia de usuario: Gestionar persona
----------------------	--

Modificación de historia de usuario: 2

Programador: Harold Pérez Trujillo	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 8 días
Riesgo en Desarrollo: Alta	Tiempo Real: 7

Descripción: muestra los campos correspondientes a registrar, listar, modificar, eliminar, detalles, buscar, eliminar seleccionadas, habilitar, deshabilitar.

Observaciones: el campo contenido está seleccionado por defecto.

Historia de usuario	
Número: HU_11	Nombre historia de usuario: Gestionar peligro
Modificación de historia de usuario: 2	
Programador: Harold Pérez Trujillo	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 9 días
Riesgo en Desarrollo: Alta	Tiempo Real: 8
Descripción: muestra los campos correspondientes a registrar, listar, modificar, eliminar, detalles, buscar, eliminar seleccionadas, habilitar, deshabilitar.	
Observaciones: el campo contenido está seleccionado por defecto.	

Historia de usuario	
Número: HU_12	Nombre historia de usuario: Gestionar riesgo
Modificación de historia de usuario: 1	
Programador: Harold Pérez Trujillo	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 9 días
Riesgo en Desarrollo: Alta	Tiempo Real: 7
Descripción: muestra los campos correspondientes a registrar, listar, modificar, eliminar, detalles, buscar, eliminar seleccionados, habilitar, deshabilitar.	
Observaciones: el campo contenido está seleccionado por defecto.	

2. Tarjetas CRC.

Nombre de la clase: Categoría de usuario	
Responsabilidades de la clase: <ul style="list-style-type: none">-Registrar una nueva categoría de usuario.-Listar todas las categorías de usuario.-Modificar una categoría de usuario.-Eliminar una categoría de usuario determinada.-Eliminar categorías de usuario seleccionadas.-Habilitar o deshabilitar una categoría de usuario.	Colaboradores: <ul style="list-style-type: none">-Usuario.

Nombre de la clase: Provincia	
Responsabilidades de la clase: <ul style="list-style-type: none">-Registrar una nueva provincia.-Listar todas las provincias registradas.-Modificar una provincia.-Eliminar una provincia determinada.-Eliminar provincias seleccionadas.-Habilitar o deshabilitar una provincia.	Colaboradores: <ul style="list-style-type: none">-Usuario.-Municipio.

Nombre de la clase: Municipio.

Responsabilidades de la clase: -Registrar un nuevo municipio. -Listar todos los municipios registrados. -Modificar un municipio. -Eliminar un municipio determinado. -Eliminar municipios seleccionados. -Habilitar o deshabilitar un municipio. -Buscar un municipio.	Colaboradores: -Entidad.
--	------------------------------------

Nombre de la clase: Entidad.	
Responsabilidades de la clase: -Registrar una nueva entidad. -Listar todas las entidades registradas. -Modificar una entidad. -Eliminar una entidad determinada. -Eliminar entidades seleccionadas. -Habilitar o deshabilitar una entidad. -Buscar una entidad.	Colaboradores: -Peligro. -Accidente laboral.

Nombre de la clase: Sector.

Responsabilidades de la clase: -Registrar un nuevo sector. -Listar todos los sectores registrados. -Modificar un sector. -Eliminar un sector determinado. -Eliminar sectores seleccionados. -Habilitar o deshabilitar un sector. -Buscar un sector.	Colaboradores: -Entidad.
---	------------------------------------

Nombre de la clase: Organismo.	
Responsabilidades de la clase: -Registrar un nuevo organismo. -Listar todos los organismos registrados. -Modificar un organismo -Eliminar un organismo determinado. -Eliminar organismos seleccionados. -Habilitar, deshabilitar y buscar un organismo .	Colaboradores: -Entidad. -Hijode.

Nombre de la clase: Accidente laboral.	
Responsabilidades de la clase: -Registrar un nuevo accidente. -Listar todos los accidentes registrados. -Modificar un accidente. -Eliminar un accidente determinado. -Eliminar accidentes seleccionados. -Habilitar o deshabilitar un accidente. -Detalles y búsqueda de un accidente .	Colaboradores: -Persona.

Nombre de la clase: Persona.	
Responsabilidades de la clase: -Registrar una nueva persona. -Listar todas las personas registradas. -Modificar una persona. -Eliminar una persona determinada. -Eliminar personas seleccionadas. -Habilitar o deshabilitar una persona. -Detalles y búsqueda de una persona.	Colaboradores:

Nombre de la clase: Peligro.	
Responsabilidades de la clase: -Registrar un nuevo peligro. -Listar todos los peligros registrados. -Modificar un peligro. -Eliminar un peligro determinado. -Eliminar peligros seleccionados. -Habilitar o deshabilitar un peligro. -Detalles y búsqueda de un peligro .	Colaboradores: -Riesgo.

Nombre de la clase: Riesgo.

<p>Responsabilidades de la clase:</p> <ul style="list-style-type: none"> -Registrar un nuevo riesgo. -Listar todos los riesgos registrados. -Modificar un riesgo. -Eliminar un riesgo determinado. -Eliminar riesgos seleccionados. -Habilitar o deshabilitar un riesgo. -Detalles y búsqueda de un riesgo. 	<p>Colaboradores:</p>
---	------------------------------

3. Pruebas Unitarias

Test Provincia

```

from django.test import TestCase
from nomencladores.models import Provincia

class ProvinciaTestCase(TestCase):
    def setUp(self):
        self.provincia = Provincia.objects.create(
            nombre='La Habana',
        )

    def test_categoria_usuario_creation(self):
        print(self.provincia.nombre)

```

Test Categoría Usuario

```

from django.test import TestCase
from nomencladores.models import CategoriaUsuario

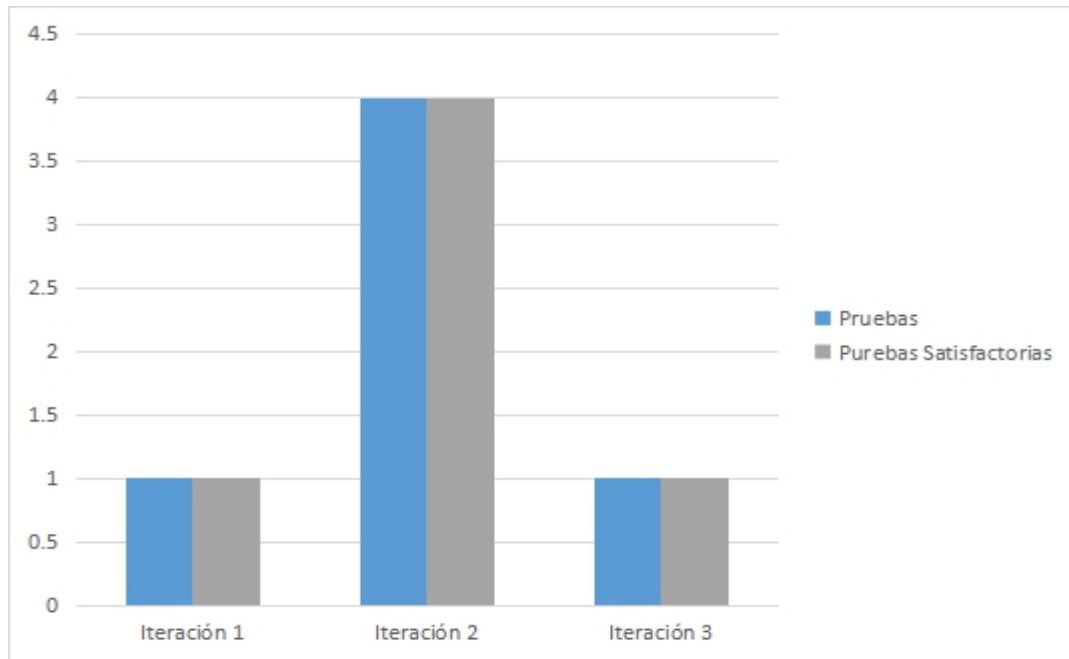
class CategoriaUsuarioTestCase(TestCase):
    def setUp(self):
        self.categoria_usuario = CategoriaUsuario.objects.create(
            categoria='Administrador',
            descripcion='Esta es la descripción',
        )

    def test_categoria_usuario_creation(self):
        print(self.categoria_usuario.categoria)
        print(self.categoria_usuario.descripcion)

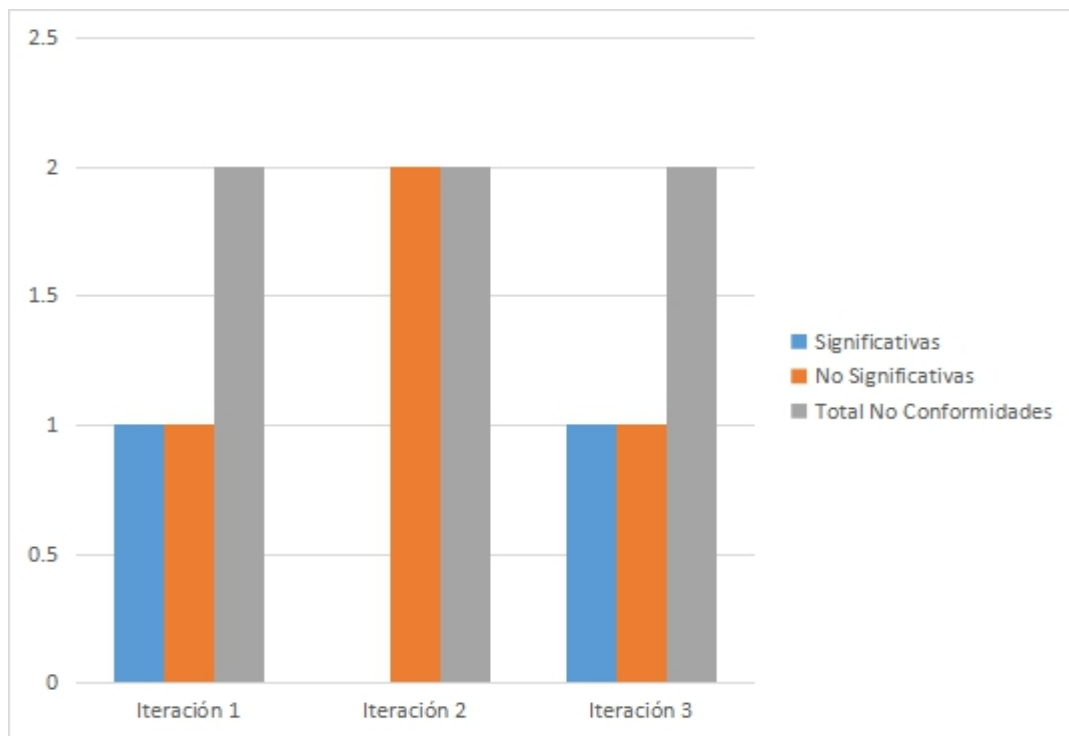
```

4. Resultados de Pruebas de Aceptación

Grado de aceptación de las Pruebas de aceptación



Estadísticas No conformidades



5. Carta de Aceptación del Cliente.

Ministerio
de Trabajo
y Seguridad
Social

La Habana 9 de diciembre de 2022
"Año 63 de la Revolución"

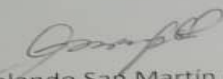
A: Universidad de Ciencias Informáticas
Ref: Cliente satisfecho

Compañero(a):

En correspondencia con la línea de referencia, tengo a bien comunicar que, como parte de la informatización del país, hemos solicitado la incorporación de educandos de las universidades a que participen de las transformaciones necesaria en nuestra esfera de competencia; en tal sentido, hemos incorporado al alumno de quinto año Harold Pérez Trujillo, quien elaboró un Software para identificar y evaluar la gestión de riesgo de la Seguridad y Salud en Trabajo.

La aplicación de esta herramienta, tendrá un alcance en toda la esfera laboral del país, su utilización propiciará evaluar con inmediatez lo que acontezca en el ambiente de trabajo, permitiendo informar oportunamente a los altos niveles de dirección.

Fraternalmente;


MSc. Angel Rolando San Martín Dupoté.
Jefe Departamento de Seguridad y Salud