



Facultad de Ciencias y Tecnologías Computacionales

Sistema Gestión de Contratos para el Centro Internacional de La Habana basado en la localización cubana de Odoo.

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

Autor(es): Xavierly Prats Ennis.

Tutor(es): Ing. Angélica Vázquez Reinoso.

La Habana, junio de 2022

Año 64 de la Revolución

DECLARACIÓN DE AUTORÍA

La autora Xavierly Prats Ennis del trabajo de diploma con título **“Sistema de Gestión de contratos para el Centro Internacional de La Habana basado en la localización cubana de Odo”** concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como única autora de su contenido. Para que así conste firman la presente a los 22 días del mes de noviembre del año 2022.

Xavierly Prats Ennis.

Firma del Autor

Ing. Angélica Vázquez Reinoso.

Firma del Tutor

AGRADECIMIENTOS

Le agradezco a mi familia sobre todo a mi abuela y mis tíos que siempre han estado pendientes de mi vida académica desde que era un piojo, a mi mamá que a pesar de que por cuestiones no hemos estado juntas físicamente siempre ha sido mi motor impulsor, a mi papá fantasma y a mis hermanos, gracias de todo corazón por todo el apoyo que me han brindado a lo largo de estos años, por siempre darme ánimos cuando he estado al borde del colapso. Gracias a mis pizzas, a mi bichito, Zoe, Kate, Kareluchy y Miguel. Gracias a todo el 106 que hicieron mi vida en la universidad un poco bastante entretenida, gracias Hassam, Jesús, Alvaringuis, Javi, Rafaela, Dayana, Adrian, Nayalis, Gilberto, Yaneysis, Sindy, Emi, Liset, Suset, Aroldo, Marco, Ariel, Frank, JuanPi, Eleandrys, Pablito, Javier, Calvo, Baberto, Arianna, mi Dayi y a Mila. Gracias a mi tutora, al tribunal, y a la oponente por tener la paciencia de explicarme las cosas. A todos ustedes Gracias.

DEDICATORIA

El presente trabajo de diploma se lo dedico a Mo chridhe, gracias a ti es que este sueño se esta haciendo posible, gracias por siempre estar junto a mí, por tirar de mi mano, por no dejarme caer nunca, por darme fuerzas, por trasnochar tantas noches conmigo estudiando hasta el cansancio, por sacrificar tus días libres enseñándome y por toda la paciencia que me tuviste.

RESUMEN

El Centro Internacional de La Habana (CIH) ofrece servicios profesionales de consultoría, transferencia de tecnología, auditoría, avalúos informáticos y de capacitación a la medida para empresas e instituciones de intereses nacionales o extranjeros. El CIH realiza su gestión de contratos a través de Microsoft Access, en este tiene tablas con toda la información de cada contrato que lleva a cabo, pero su proceso de contratación se realiza de forma lenta y engorrosa. El presente trabajo tiene como objetivo desarrollar un sistema para la gestión de los contratos que aumente el grado de integralidad y eficiencia en esta empresa. Para el desarrollo de la solución se utilizó como guía ingenieril la metodología SCRUM. La verificación y validación del correcto funcionamiento de la propuesta de solución se realizó mediante pruebas, obteniéndose resultados satisfactorios en cuanto a su utilidad.

PALABRAS CLAVE

Sistema, Contratos, Gestión, optimizar.

ABSTRACT

The Havana International Center (CIH) offers professional consulting services, technology transfer, auditing, computer appraisals, and tailor-made training for companies and

institutions of national or foreign interest. The CIH carries out its contract management through Microsoft Access, in which it has tables with all the information of each contract it carries out, but its contracting process is carried out slowly and cumbersome. The objective of this work is to develop a system for the management of contracts that increases the degree of integrity and efficiency in this company. For the development of the solution, the SCRUM methodology was used as an engineering guide. The verification and validation of the correct functioning of the solution proposal was carried out through tests, obtaining satisfactory results in terms of its usefulness.

KEYWORDS

System, management, contract, optimize.

Índice

INTRODUCCIÓN.....	12
CAPÍTULO 1: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS SISTEMAS DE GESTIÓN DE CONTRATOS.....	17
1.1 Conceptos fundamentales.....	17
1.2 Estado del arte de los sistemas gestión de contratos.....	19
1.2.1 Soluciones Nacionales	19

1.2.2 Soluciones Internacionales:	20
1.2.3 Comparación entre los sistemas	21
1.3 Metodología.....	22
1.4 Lenguajes, Tecnologías y herramientas:.....	23
Conclusiones del capítulo.....	27
CAPÍTULO 2: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE CONTRATOS.	
.....	28
2.1 Modelo Conceptual.....	28
2.2 <i>Product Backlog</i> :.....	29
2.2.1 Requisitos del sistema	29
2.3 Planificación del <i>Sprint Backlog</i> :.....	34
2.4 Historias de Usuario.....	35
2.5 Arquitectura.....	37
2.6 Diagrama de clases de diseño.....	39
Descripción de los elementos del Diagrama:.....	40
2.7 Patrones de diseño.....	41
2.8 Modelo Entidad Relación.....	42
2.9 Diagrama de Despliegue:.....	43
Conclusiones del capítulo.....	45
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	46
3.1 Pruebas de software.....	46
3.2 Estrategia de pruebas de <i>software</i>	46
3.2.1 Aplicación de las pruebas	47
Conclusiones del capítulo.....	57
CONCLUSIONES FINALES.....	58
RECOMENDACIONES.....	59
REFERENCIAS.....	60
ANEXOS.....	62

ÍNDICE DE TABLAS

Tabla 1 Comparación de soluciones informáticas (elaboración propia).....	21
Tabla 2 Descripción de los requisitos funcionales.....	31
Tabla 3 Sprint 0.....	34
Tabla 4 Sprint 1.....	35
Tabla 5 Historia de Usuario del requisito Crear contrato.....	35
Tabla 6 Historia de Usuario del requisito Editar empleado.....	36
Tabla 7 Historia de Usuario del requisito Autenticar usuario.....	36
Tabla 8 Partición de equivalencia del caso de prueba “Crear contrato”. (Elaboración propia).....	49
Tabla 10 Camino básico.....	54
Tabla 11 Descripción del camino básico.....	54

ÍNDICE DE FIGURAS

Figura 1 Modelo conceptual.....	29
Figura 2 Arquitectura Modelo Vista Controlador.....	38
Figura 3 Diagrama de paquetes (Elaboración Propia).....	38
Figura 4 Diagrama de Clases del Diseño Gestionar Contrato.....	40
Figura 5 Patrón Controlador.....	41
Figura 6 Patrón Experto.....	42
Figura 7 Modelo Entidad Relación de la Base de datos.....	43
Figura 8 Diagrama de Despliegue.....	43
Figura 9 Resultados de la prueba de caja negra (Elaboración laboral).....	53
Figura 10 Método onchange_company_id.....	53
Figura 11 Captura de pantalla del sistema (Autenticar usuario).....	55
Figura 12 Captura de pantalla del sistema (Crear contrato).....	56

INTRODUCCIÓN

El desarrollo de las Tecnologías de Información y Comunicaciones (TIC) y su repercusión en todas las esferas de la vida moderna, ha logrado que el sector empresarial opte por incorporar sistemas de gestión para automatizar las tareas que se desarrollan en cada área de la empresa. Las TIC permiten la obtención de un mayor volumen de información en términos más precisos y exactos, lo que hace posible que las empresas ofrezcan productos y servicios con un mayor grado de adaptación a las características de los compradores, y no sólo a través de establecimientos físicos.

En las empresas la gestión comercial es un término relacionado a la mercadotecnia y se encarga de administrar y gestionar adecuadamente las principales actividades comerciales de las organizaciones en el mercado.[CITATION Mir18 \l 3082]

La comercialización de bienes y servicios requiere del establecimiento de contratos entre partes, que permitan alcanzar resultados satisfactorios. Un contrato, es una categoría que jurídicamente establece lo que acontece en el ámbito económico, al facilitar el intercambio de bienes y servicios. La gestión de contratos se encarga de que el producto o servicio que se solicita, se entrega o presta en el tiempo y con la calidad establecida. También permite que dicho proceso garantice que tanto el proveedor como el cliente actuarán acorde a los requisitos contractuales.

En Cuba se han creado instituciones y empresas que desarrollan soluciones informáticas para la automatización de procesos en todas las esferas de la sociedad, como ejemplo de este tipo de instituciones se hace referencia a: La Universidad de las Ciencias Informáticas (UCI), la cual desde su misión proyecta el desarrollo de productos y servicios que sirvan de soporte a la industria del software en el país.

El Centro Internacional de La Habana (CIH) es una sociedad mercantil que ofrece servicios profesionales de excelencia en Cuba y en el extranjero, atesorado por la experticia de sus profesionales y el conocimiento existente en las Universidades y Entidades, Tecnología e

Innovación, brindando soluciones integrales y apostando por el desarrollo sostenible.
[CITATION CIH22 \l 3082]

El proceso de contratación en esta sociedad mercantil se realiza manualmente y una vez que ambas partes están de acuerdo con el contrato establecido se procede a la firma y se crea una copia de dicho contrato almacenado en un sistema gestor de base de datos, Access, este proceso es un poco lento pues no es un sistema conectado que permita tener información inmediata, además trae consigo riesgos de pérdida de información de los contratos e información del cliente.

A partir de la descripción del proceso anterior se identificaron las siguientes limitaciones que dan origen a la situación problemática:

- ✓ El tiempo para la negociación, creación, aprobación y renovación de contratos no cumple con las expectativas del cliente.
- ✓ Existe pérdida de información del contrato y datos del cliente ya que este procedimiento se realiza de forma manual por lo que no se logra registrar la totalidad de los datos.

Dada la situación problemática antes descrita, se plantea como **problema de investigación**: ¿Cómo lograr una correcta gestión de los contratos en el Centro Internacional de La Habana?

Este problema está centrado en el **objeto de estudio**: Sistema para la gestión de contratos.

Se define como **campo de acción**: Desarrollo de un sistema para la gestión de contratos en el Centro Internacional de La Habana.

Para dar solución al problema antes mencionado, se traza como **objetivo general** de la investigación: Desarrollar un sistema para la gestión de los contratos que se realizan en el Centro Internacional de La Habana.

Para apoyar el cumplimiento del objetivo general se tendrán en cuenta los siguientes objetivos específicos:

1. Analizar los fundamentos teóricos para el desarrollo de soluciones de *software* para gestión de Contratos.
2. Diseñar e implementar la solución de *software*.
3. Validar a través de pruebas funcionales la solución desarrollada.

Para dar solución al problema de la investigación y a los objetivos específicos formulados, se proponen las siguientes **tareas de investigación**:

- ✓ Elaboración de un marco teórico referencial sobre los sistemas de gestión para el encuadre correcto de la investigación.
- ✓ Estudio diagnóstico sobre los sistemas de gestión para la gestión de contratos.
- ✓ Diseño de un sistema de gestión para la gestión de contratos del Centro Internacional de La Habana.
- ✓ Validación de la gestión de contratos con la solución propuesta a partir de los resultados obtenidos, mediante pruebas para evaluar el correcto funcionamiento del sistema.

Con el desarrollo de la presente investigación se obtendrá como **posibles resultados**: una solución informática, que permitirá, una ágil gestión de contratos en el Centro Internacional de La Habana y artefactos de ingeniería para la documentación del desarrollo.

Para darle cumplimiento a las tareas de investigación se emplearán varios **métodos científicos** en la búsqueda y procesamiento de los datos tales como:

Métodos teóricos:

- ✓ **Histórico-Lógico:** permitió realizar un estudio sobre las principales tendencias con respecto a los sistemas de gestión comercial ya existentes, tanto en el mundo, como en el marco productivo del país, con el fin de seleccionar las más apropiadas para el desarrollo del sistema.

Métodos empíricos:

- ✓ **La entrevista:** se utilizó a la hora del análisis del negocio, con el objetivo de lograr una visión más amplia del mismo y llegar con mayor eficiencia a los requisitos necesarios para la implementación del sistema.
- ✓ **Análisis documental:** Este método se aplicó para extraer la información necesaria de la bibliografía existente y consultar información en sitios de interés nacional e internacional, con el objetivo de apoyar las tareas de investigación.

El contenido de la presente investigación se encuentra estructurado en los siguientes tres capítulos:

Capítulo 1: Fundamentos y referentes teóricos-metodológicos sobre los sistemas de Gestión de Contratos.

Este capítulo contiene los fundamentos teóricos que sustentan al sistema de gestión de contratos. En el mismo se hace un estudio de sistemas homólogos analizando el objetivo, su funcionamiento y tecnologías que utilizan. Además, se describen los principales conceptos para comprender el dominio del problema, así como las tendencias, metodologías, tecnologías y herramientas que se empleará en la implementación de la propuesta de solución.

Capítulo 2: Diseño e implementación del sistema de Gestión de contratos.

En este capítulo se explica cómo se desarrolla el flujo actual de los procesos y se describe la propuesta de solución para resolver el problema planteado. Se especifican los requisitos

funcionales y no funcionales y los elementos fundamentales del diseño y de la arquitectura que se deben tener en cuenta para el desarrollo del sistema, asistido por la realización de diagramas.

Capítulo 3: Validación del sistema.

En este capítulo se detalla la propuesta solución al problema planteado. Se define la estrategia de pruebas definidas para validar la propuesta solución verificando el cumplimiento de los requisitos funcionales y no funcionales descritos en el capítulo anterior y se muestran imágenes de interfaz del sistema como parte del resultado final.

CAPÍTULO 1: Fundamentos y referentes teórico-metodológicos sobre los sistemas de gestión de contratos.

En el presente capítulo se define el fundamento teórico de la investigación, exponiéndose los principales conceptos relacionados con la gestión de contratos, para una mejor comprensión del tema. Se describe además la metodología escogida para guiar el proceso de desarrollo de *software*, así como los lenguajes, herramientas y tecnologías que se emplearán para el desarrollo del sistema.

1.1 Conceptos fundamentales.

A continuación, se exponen los principales conceptos relacionados con el marco teórico de la investigación, con el objetivo de profundizar en los distintos puntos de vista y asumir una posición al respecto.

Contrato:

Según Ignacio Galindo Garfias: “un contrato el acto jurídico típico del derecho privado y se caracteriza porque las declaraciones de voluntad de las partes que lo celebran son concurrentes, convienen en crear entre sí relaciones jurídicas; forman un acuerdo de voluntades o consentimiento, lo que constituye propiamente el elemento esencial del contrato. El contrato puede ser considerado como acto jurídico por una parte y como norma individualizada de conducta desde otro punto de vista”. [CITATION Lia17 \l 3082]

Azúa Reyes define que: “un contrato es el acuerdo de voluntades de dos o más personas para crear o transferir entre sí, derechos y obligaciones. De su naturaleza explica que es un acto jurídico bilateral o plurilateral, o si se prefiere, dos o más actos jurídicos emanados respectivamente de dos o más personas que concurren a la realización de un negocio en el que se encuentran interesadas”. [CITATION Lia17 \l 3082]

Rafael de Pina: “en su significación semántica, contrato es el pacto o convenio entre partes sobre materia o cosa determinada y a cuyo cumplimiento pueden ser exigidas. El contrato no es una norma jurídica en el mismo sentido en que lo son la ley y la costumbre, por ejemplo, sino un acto jurídico celebrado de acuerdo con determinadas normas jurídicas”. [CITATION Lia17 \l 3082]

A partir de las definiciones anteriores se puede definir que un contrato es un acuerdo escrito o documento electrónico entre dos o más personas para establecer el cumplimiento de ciertas normas entre los negociantes.

Contrato electrónico: En sentido estricto, es aquel contrato que se perfecciona mediante un intercambio electrónico de datos de ordenador a ordenador. Frente a esta noción, existe una amplia que incluye dentro de dicha categoría todos los contratos celebrados por medios electrónicos (aunque no se hayan utilizado ordenadores), como fax, télex y teléfonos.
[CITATION Jor98 \l 3082]

Gestión de Contratos: Es la actividad que busca administrar de manera eficaz un acuerdo que surge entre dos partes para alcanzar un beneficio para ambos, se entiende como una disciplina que forma parte de la gestión de proyectos y administra eficientemente las obligaciones que dos o más partes contrajeron para desarrollar un servicio exitosamente.
[CITATION Cés18 \l 3082]

Sistema de Gestión Comercial (CGM): Son un enfoque estratégico de gestionar el valor y las experiencias con los clientes. Su tecnología es un facilitador de los beneficios que llegarán de aplicar una estrategia comercial adecuada, una gran capacidad analítica y unas políticas innovadoras.[CITATION Kal14 \l 3082]

Sistema Gestión (SG): Es una serie de procesos, acciones y tareas que se realizan sobre un conjunto de elementos (personas, procedimientos, estrategias, planes, recursos y productos) para lograr el éxito sostenido de una organización, disponer de capacidad para satisfacer las necesidades y las expectativas de sus clientes o beneficiarios, trabajadores y de otras partes interesadas a largo plazo y de un modo equilibrado y sostenible.[CITATION Fra15 \l 3082]

Un SG no solamente está indicado para grandes empresas. Permite a la pequeña y mediana empresa aprovechar sus características para conseguir un perfecto conocimiento del funcionamiento interno, detectando los posibles puntos débiles en la gestión y facilitando la labor de corrección de los mismos. Un SG es una herramienta fundamental para conseguir unos elevados niveles de control y calidad.[CITATION Cri16 \l 3082]

Sistema web: se denomina sistema web a aquellos sistemas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador.

Por lo antes expuesto se concluye, que un sistema de gestión de contratos es una serie de procesos que administran la forma eficaz de llevar a cabo convenios entre varias personas en un corto periodo de tiempo.

1.2 Estado del arte de los sistemas gestión de contratos.

Existen actualmente diferentes sistemas informáticos que sirven de apoyo a los procesos de gestión de contratos. Para un mejor entendimiento de las funcionalidades a desarrollar en la solución, es necesario, realizar un estudio de sistemas que poseen características similares los cuales se describen a continuación.

1.2.1 Soluciones Nacionales

ERP Zoom Localización Cubana de Odoo está desarrollado sobre la versión comunitaria del ERP belga Odoo, tecnología open source con más de 3 millones de usuarios y una amplia comunidad internacional de respaldo. El sistema, adaptado a las normas y regulaciones nacionales fundamentalmente en materia contable financiera, integra más de 30 módulos o aplicaciones para la gestión de diferentes áreas de un negocio. [CITATION Des22 \l 3082]

Permite a las entidades cubanas integrar en un sistema único todos los procesos asociados con los aspectos estratégicos, operativos o productivos. Facilita e integra la información entre las funciones de manufactura, logística, finanzas, recursos humanos, mercadotecnia, etc. De igual manera, administra y fomenta las relaciones con clientes y proveedores, además de habilitar canales de comunicación y gestionar las interacciones al interior de cada proceso.

Su estructura permite cubrir la necesidad de información de cada usuario de manera independiente, proporcionando tableros de mando y reportes personalizados que permiten la autogestión del control interno, la evaluación y el seguimiento a resultados por actividad y la toma de decisiones de una manera más eficaz y efectiva. Enfocado a proporcionar soporte a las empresas para explotar todas sus potencialidades, cuenta en su estructura modular con

herramientas para el desarrollo del comercio electrónico, la promoción de productos y servicios y el *marketing* digital.

1.2.2 Soluciones Internacionales:

Juro:

Juro es una solución de gestión de contratos de extremo a extremo que apoya la creación de contratos, negociación, firma electrónica y analíticas para acuerdos de no divulgación, contratos de empleo, contratos de ventas, acuerdos de nivel de servicio, contratos de SaaS empresarial y mucho más. Como solución en la nube, *Juro* permite a los usuarios gestionar todo el ciclo de vida del contrato cómodamente en línea a través de cualquier dispositivo habilitado para internet, incluidos portátiles, tabletas, móviles y de escritorio. Con una API escalable, *Juro* ayuda a los usuarios a agilizar su flujo de trabajo [CITATION Jur09 \l 3082].

Juro proporciona una gama de funciones automatizadas diseñadas para ayudar a los usuarios a realizar un seguimiento y gestionar la actividad de los contratos, como plantillas automatizadas, alertas y controles de acceso. Las alertas garantizan que los usuarios nunca pierdan los plazos de renovación, mientras que los controles de acceso garantizan que solo las personas autorizadas tengan derecho a acceder a la información del contrato. *Juro* admite la recolección y el análisis de datos, lo que proporciona a los usuarios una visibilidad completa de los datos y el rendimiento del contrato. Las pistas de comentarios en *Juro* permiten a los usuarios discutir las versiones de contratos dentro de la misma plataforma, lo que permite ahorrar tiempo en el envío y la devolución de correos electrónicos. [CITATION Jur09 \l 3082]

- ✓ Funcionalidades que brinda:

- ✓ Automatización de documentos.

- ✓ Gestión de contratos.

- ✓ Gestión de contratos o licencias.

- ✓ Gestión de documentos.

- ✓ Gestión de flujo de trabajo.
- ✓ Gestión de tareas.
- ✓ Búsqueda de texto completo.
- ✓ Creación de comentarios/notas.
- ✓ Creación de informes y estadísticas.

ContractWorks:

ContractWorks es una solución de gestión de contratos simple e inteligente que te permite obtener el control total de tus contratos, [facilitando la ejecución, el almacenamiento y el seguimiento de los acuerdos corporativos](#), de manera asequible y rápida, con usuarios ilimitados. *ContractWorks* presenta una implementación rápida, una interfaz fácil de usar y precios bajos y transparentes. *ContractWorks* hace que el *software* de gestión de contratos sea accesible para empresas y equipos de todos los tamaños. Además de las funciones necesarias para gestionar los acuerdos, obtendrás un equipo de profesionales de asistencia dedicados para ayudar a tu empresa a aprovechar al máximo tu portfolio de contratos. Permitiendo una serie de funcionalidades entre las que se incluyen:

- ✓ Gestión de auditorías.
- ✓ Gestión de contratos o licencias.
- ✓ Gestión de documentos.
- ✓ Gestión de la conformidad.
- ✓ Gestión del ciclo de vida de los contratos.

- ✓ La importación y exportación de datos.
- ✓ Almacenamiento de documentos.
- ✓ Creación de informes/análisis.
- ✓ Extracción de datos.

Volody Contract LifeCycle Management

Volody Contract LifeCycle Management es una herramienta inteligente habilitada para la Inteligencia artificial (IA) y *Machine learning* (ML aprendizaje automático, por sus siglas en inglés) que proporciona una experiencia de gestión de contratos de alto nivel. Con *Volody*, puedes crear, realizar un seguimiento, colaborar y obtener información sobre todos los procesos relacionados con contratos. Es una gran solución para liberar el potencial de la automatización de contratos.[CITATION Vol17 \l 3082]

Las funcionalidades del sistema cuentan con:

- ✓ Gestión de auditorías.
- ✓ Notificación en tiempo real.
- ✓ Gestión de contratos.
- ✓ Gestión de documentos.
- ✓ Gestión de modelos.
- ✓ Control de versiones.
- ✓ Gestión del ciclo de vida de los contratos.

- ✓ Firma digital.
- ✓ Firma electrónica.
- ✓ Almacenamiento de documentos.
- ✓ Creación de informes/análisis.
- ✓ Extracción de datos.

1.2.3 Comparación entre los sistemas

Tabla 1 Comparación de soluciones informáticas (elaboración propia).

Solución Informática	Software Libre	Gratuito	Soporte web	Disponible para utilizar en Cuba	Gestión de Contratos
Localización cubana basada en Odo	Sí	Sí	Sí	Sí	Sí
<i>Juro</i>	No	No	No	No	Sí
<i>ContractWorks</i>	No	No	No	No	Sí
<i>Volody Contract LifeCycle Management</i>	No	No	Sí	No	Sí

Valoración de las soluciones existentes:

Después de analizar cada uno de los sistemas antes mencionadas como posible solución, se puede decir, que estos son sistemas que pueden cumplir con los la problemática planteada, pero al ser estos de pago y no cumplen con la política del país de software libre, como se puede observar en la tabla comparativa antes planteada (ver Tabla 1), por tanto, no es posible su utilización para cumplir con la solución planteada. Por lo que se tuvo en cuenta para la solución, la localización cubana basada en Odo, la cual, cuenta con un módulo de gestión de contratos, esta solución informática es una *Application Programming Interface* (API) que puede ser utilizada para el desarrollo del sistema de gestión pues cumple con la política establecida por el país de software libre, por tanto, es posible desarrollar el sistema basado en esta solución existente, evitando así generar gastos adicionales al Centro Internacional de La Habana.

1.3 Metodología.

El desarrollo de un proyecto de *software* es un proceso que engloba procedimientos, técnicas, documentación y herramientas que se utilizan en la creación de un producto de *software*. Convirtiéndose dicha metodología en un plano que apoya a los desarrolladores, para así lograr que el cliente quede satisfecho con el resultado final.

La metodología Scrum para el desarrollo ágil de *software* es un marco de trabajo diseñado para lograr la colaboración eficaz de equipos en proyectos, que emplea un conjunto de reglas y artefactos y define roles que generan la estructura necesaria para su correcto funcionamiento.

Scrum utiliza un enfoque incremental que tiene como fundamento la teoría de control empírico de procesos. Esta teoría se fundamenta en transparencia, inspección y adaptación; la transparencia, que garantiza la visibilidad en el proceso de las cosas que pueden afectar el resultado; la inspección, que ayuda a detectar variaciones indeseables en el proceso y la adaptación, que realiza los ajustes pertinentes para minimizar el impacto de las mismas. [CITATION Nav13 \l 3082]

Roles de Scrum:

- ✓ **Scrum Máster:** El *Scrum master* tiene como función asegurar que el equipo está adoptando la metodología, sus prácticas, valores y normas; es el líder del equipo, pero no gestiona el desarrollo.
- ✓ **Product Owner:** Representa a los interesados, es el responsable de maximizar el valor del producto y el trabajo del equipo de desarrollo; tiene entre sus funciones gestionar la lista ordenada de funcionalidades requeridas o *Product Backlog*.
- ✓ **Equipo de desarrollo:** Está integrado por los desarrolladores, administradores de base de datos, *testers*.

Generación de artefactos:

- ✓ *Product Backlog*: Es el listado de requisitos funcionales y la descripción de los mismos.
- ✓ *Sprint*: Corresponde a una ventana de tiempo donde se crea una versión utilizable del producto (incremento). Cada *Sprint*, como en el *rugby*, es considerado como un proyecto independiente. Su duración máxima es de un mes.
- ✓ *Sprint Backlog*: Es la planificación que se realiza para saber qué requisitos se implementan en cada *sprint*.
- ✓ Historias de usuario: Scrum utiliza las historias de usuario para agrupar y dar tratamiento al modelado de los requisitos establecidos en el *Product Backlog*.

Los elementos de un sprint en Scrum son:

- ✓ *Daily Scrum*: es un evento del equipo de desarrollo de quince minutos, que se realiza cada día con el fin de explicar lo que se ha alcanzado desde la última reunión; lo que se hará antes de la siguiente; y los obstáculos que se han presentado.
- ✓ La Revisión del *Sprint*: ocurre al final del *Sprint* y su duración es de cuatro horas para un proyecto de un mes (o una proporción de ese tiempo si la duración es menor).
- ✓ La Retrospectiva del *Sprint*: es una reunión de tres horas del equipo Scrum en la que se analiza cómo fue la comunicación, el proceso y las herramientas; qué estuvo bien, qué no, y se crea un plan de mejoras para el siguiente *Sprint*.

1.4 Lenguajes, Tecnologías y herramientas:

HTML 5: *El HyperText Markup Language (HTML)* es un lenguaje de marcas hipertextuales, un lenguaje diseñado para estructurar textos para generar páginas web. La mayoría de las etiquetas del lenguaje HTML son semánticas. La interpretación de las etiquetas es realizada por el navegador web. El lenguaje HTML es extensible, se le pueden añadir características,

etiquetas y funciones adicionales para el diseño de páginas web, generando un producto vistoso, rápido y sencillo.[CITATION Jua12 \l 3082]

CSS 3: *Cascading Style Sheets* (Hojas de Estilo en Cascada), es un mecanismo que describe cómo se va a mostrar un documento en la web, o incluso cómo se va a imprimir. Este mecanismo ofrece a los desarrolladores el control sobre el estilo y el formato de sus documentos. Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Estos estilos permiten a los desarrolladores web controlar el formato de múltiples páginas web al mismo tiempo, posibilitando ante cualquier cambio en el estilo marcado para un elemento en la CSS, que sean afectadas todas las páginas vinculadas a dicha CSS en las que aparezca ese elemento.[CITATION Jua12 \l 3082]

Entre los beneficios concretos de CSS se encuentran:

- ✓ Control de la presentación de muchos documentos desde una única hoja de estilo.
- ✓ Control más preciso de la presentación.
- ✓ Aplicación de diferentes presentaciones a diferentes tipos de medios.
- ✓ Numerosas técnicas avanzadas y sofisticadas

Bootstrap v 5.1: Bootstrap es un *framework* CSS de código abierto. Se caracteriza por ser una excelente herramienta para crear interfaces de usuarios totalmente adaptables a cualquier tipo de dispositivo y pantalla, independientemente de su tamaño. Además, es compatible con la mayoría de navegadores web del mercado como: *Google Chrome, Safari, Mozilla Firefox, Internet Explorer* y *Opera*. [CITATION Jua11 \l 3082]

Java Script 1.8.5: *JavaScript* es el lenguaje que nos permite interactuar con el navegador de manera dinámica y eficaz, proporcionando a las páginas web rapidez y vida. *JavaScript* comparte muchos elementos con otros lenguajes de alto nivel. Hay que tener en cuenta que este lenguaje es muy semejante a otros como C, *Java* o *PHP*, tanto en su formato como en su sintaxis, aunque tiene sus propias características definitorias.[CITATION Lun19 \l 3082]

Python 3.9.5: *Python* es un lenguaje de programación multiparadigma, pues soporta orientación a objetos, programación imperativa, programación orientada a aspectos y en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico, es fuertemente tipado y multiplataforma.[CITATION Jon18 \l 3082]

Lenguaje Unificado de Modelado 2.0 (UML, por sus siglas en inglés): es el lenguaje de modelado de sistemas de *software* más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables[CITATION Car12 \l 3082]

Visual Paradigm para UML 8.0: Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. También ayuda de forma rápida a la construcción de aplicaciones de calidad, permite dibujar todo tipo de diagramas de clases y generar documentación. La herramienta proporciona abundantes tutoriales de UML, posee una interfaz sugerente, pertenece a la familia de *software* libre y soporta UML como notación principal.[CITATION Rah22 \l 3082]

API: Las siglas API vienen del inglés *Application Programming Interface* (Interfaz de Programación de Aplicaciones), es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro *software*. [CITATION Hol08 \l 3082]

JSON: Es un formato de texto sencillo para el intercambio de datos.[CITATION Rub17 \l 3082]

Ajax: Es el acrónimo de *Asynchronous JavaScript and XML*, es una biblioteca basada en *JavaScript* que permite hacer una petición al servidor sin tener que renderizar todo el archivo HTML, utiliza el formato JSON de la web para devolver las respuestas de las solicitudes clientes al servidor.[CITATION Alb17 \l 3082]

Sistema Gestor de Base de Datos PostgreSQL 6.8: Un sistema gestor de base de datos (SGBD) es el programa o conjunto de programas que gestionan y mantienen consistentes los datos almacenados en la base de datos. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.[CITATION Bac15 \l 3082]

PostgreSQL es un sistema de gestión de bases de datos relacional orientada a objetos y de software libre. Entre sus características se destacan:

- ✓ Gran escalabilidad. Capaz de ajustarse al número de Unidades Centrales de Procesamiento (del inglés *Central Processing Units CPU*) y a la cantidad de memoria que posee el sistema de forma óptima.
- ✓ Manejo de gran volumen de datos.
- ✓ Ofrece la posibilidad de ejecutar y trabajar varios procesos al mismo tiempo sobre la misma tabla sin ser dañada.
- ✓ Tiene gran soporte para vistas, procedimientos ubicados en el servidor, transacciones y características orientadas a objetos.

Se define esta herramienta para el desarrollo de la propuesta de solución debido a la estabilidad, potencia, robustez y facilidad de administración que provee.

Conclusiones del capítulo

Después de realizar el estudio de la bibliografía relacionada con los sistemas para la gestión de contratos, se pudo identificar que las soluciones existentes no cumplen con la política del país de software libre. Por esta razón se propone la implementación de un sistema que sea capaz de ajustarse a las necesidades existentes y que se nutra de las principales ventajas que muestran algunas de las soluciones analizadas como por ejemplo la capacidad de ser un sistema de gestión, que cumpla con la política de *software* libre que desea tener el país y permita la generación de contratos.

Del análisis sobre las tendencias actuales en el desarrollo del *software*, se seleccionó la metodología de desarrollo y un grupo de tecnologías y herramientas, las adecuadas para realizar el sistema informático que se requiere: como metodología de desarrollo, se escogió la metodología ágil Scrum, como sistema de gestión de base de datos *PostgreSQL* 6.8 Como lenguaje de modelado UML 2.0 y *Visual Paradigm* para UML en su versión 8.0, como lenguaje de programación se seleccionó *JavaScript* v1.8.5 y *CSS* 3, junto con el lenguaje de etiqueta *HTML* 5 para la programación desde el lado del cliente, se empleará además para el consumo de la API (Localización cubana basada en *Odoo*), la tecnología *Ajax* para manejar las peticiones del servidor conjuntamente con el formato de intercambio de datos *JSON*.

CAPÍTULO 2: Diseño e implementación del Sistema de Gestión de Contratos.

En el presente capítulo se describen las principales características de la propuesta de solución y se identifican los requisitos funcionales y no funcionales. Se describe la arquitectura de software, los patrones de diseños y los artefactos generados para darle solución al problema planteado en la investigación.

2.1 Modelo Conceptual

Un modelo conceptual es un artefacto de la disciplina de análisis, construido con las reglas UML. Tiene como objetivo comprender y describir las clases más importantes, así como, identificar y explicar los conceptos significativos en el dominio del problema, identificando los atributos y las asociaciones existentes entre ellos. Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Se realiza cuando no se logra determinar el proceso del negocio con fronteras bien establecidas y donde los flujos de información son difusos (múltiples orígenes, sólo eventos, sucesos), cuando existe solapamiento de responsabilidades, así como múltiples responsabilidades. Se representa con un conjunto de diagramas de clases UML en los que no se define ninguna operación. [CITATION Cra05 \l 3082]

El Centro Internacional de La Habana solicita la realización de un sistema de gestión de contratos para utilizarlo en su organización, donde el empleado realiza una solicitud de contrato la cual puede ser aprobada o no por el ejecutor, cuando ambas partes están de acuerdo se procede a la firma del contrato.

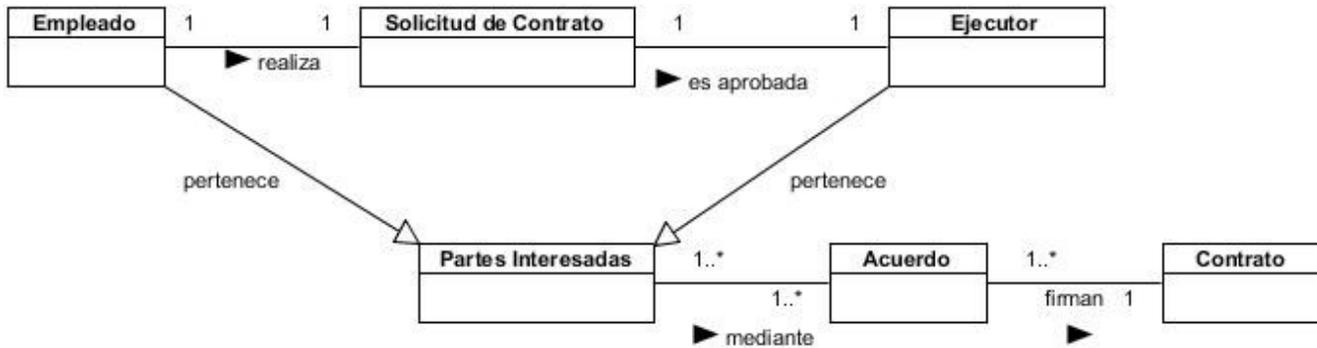


Figura 1 Modelo conceptual

A continuación, se muestra una breve descripción de los conceptos asociados al modelo conceptual:

- ✓ **Empleado:** El empleado es la persona que se beneficiará del contrato.
- ✓ **Ejecutor:** Es la persona encargada de reunirse con el empleado y llegar a un acuerdo con el mismo.
- ✓ **Acuerdo:** Representa que ambas partes están conformes con las actividades que se realizarán.
- ✓ **Solicitud de contrato:** Planilla con los datos del empleado y sus requerimientos.
- ✓ **Partes Interesadas:** Representa al empleado y al ejecutor que son las personas que intervienen en la firma del contrato.

2.2 Product Backlog:

2.2.1 Requisitos del sistema

Los requisitos se pueden clasificar en funcionales y no funcionales. Pueden documentar las interfaces externas, describir la funcionalidad y el rendimiento del sistema.

Requisitos funcionales:

Los requisitos funcionales (RF) describen lo que el sistema debe hacer. Estos dependen del tipo de *software* que se desarrolle, los posibles usuarios y el enfoque tomado por el proyecto. Además, describen con detalle la función del sistema, sus entradas, excepciones y

salidas[CITATION Som111 \l 3082]. A continuación, se describen los requisitos funcionales del sistema:

RF1 Autenticar usuario

RF2 Registrar usuario

RF3 Editar usuario

RF4 Eliminar usuario

RF5 Visualizar lista de usuarios

RF6 Crear contrato

RF7 Editar contrato

RF8 Buscar contrato

RF9 Eliminar contrato

RF10 Visualizar lista de contratos

RF11 Crear ejecutor

RF12 Editar ejecutor

RF13 Visualizar lista de ejecutores

RF14 Eliminar ejecutor

RF15 Crear dirección

RF16 Editar dirección

RF17 Visualizar lista de direcciones

RF18 Eliminar dirección

RF19 Crear empleado

RF20 Editar empleado

RF21 Visualizar lista de empleados

RF22 Eliminar empleado

RF23 Crear cliente

RF24 Editar cliente

RF25 Buscar cliente

RF26 Eliminar cliente

Tabla 2 Descripción de los requisitos funcionales.

ID	Nombre del requisito	Descripción	Prioridad
1	Autenticar usuario	El sistema permite que el usuario introduzca sus datos para acceder al sitio web.	Media
2	Registrar usuario	El sistema permite registrar usuarios.	Media
3	Editar usuario	El sistema permite editar los datos de un usuario.	Media
4	Eliminar usuario	El sistema permite eliminar los datos de un usuario.	Media
5	Visualizar lista de usuarios	El sistema permite ver la lista de los usuarios que se encuentran registrados.	Media
6	Crear contrato	El sistema permite crear contratos.	Alta
7	Editar contrato	El sistema permite editar los datos de un contrato.	Alta
8	Buscar contrato	El sistema permite buscar contratos.	Alta
9	Eliminar contrato	El sistema permite que el empleado/ejecutor del sistema pueda eliminar el contrato.	Alta
10	Visualizar lista de contratos	El sistema permite que el ejecutor vea el listado de los contratos.	Alta.
11	Crear ejecutor.	El sistema permite registrar el ejecutor de cada contrato.	Alta
12	Editar ejecutor.	El sistema permite modificar el ejecutor de cada contrato	Alta
13	Visualizar lista de ejecutores.	El sistema permite listar a todos los ejecutores	Alta.

		registrados.	
14	Eliminar ejecutor.	El sistema permite que el administrador del sistema pueda eliminar los ejecutores.	Alta.
15	Crear dirección	El sistema permite insertar la dirección de los contratos.	Alta.
16	Editar dirección	El sistema permite editar la dirección de los contratos.	Alta
17	Visualizar lista de direcciones	El sistema permite ver la lista direcciones de los contratos.	Alta
18	Eliminar dirección	El sistema permite eliminar direcciones.	Alta
19	Crear empleado	El sistema permite crear empleados.	Alta
20	Editar empleado	El sistema permite editar los datos de los empleados.	Alta
21	Visualizar lista de empleados	El sistema permite listar los empleados que se han registrado en el sistema.	Alta
22	Eliminar empleado	El sistema permite eliminar los empleados que han sido registrados.	Alta
23	Crear cliente	El sistema permite crear clientes.	Alta
24	Editar cliente	El sistema permite editar el cliente creado.	Alta
25	Buscar cliente	El sistema permite buscar un cliente.	Alta
26	Eliminar cliente	El sistema permite eliminar clientes.	Alta

Requisitos no funcionales:

Los requisitos no funcionales son propiedades que hacen al producto atractivo, usable, rápido o confiable. Se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el *software*. [CITATION Som11 \l 3082]

Software:

RNF1: Tener un navegador instalado que sea compatible con CSS3 y HTML5.

Usabilidad:

RNF2: Los mensajes para interactuar con los usuarios y los de error deben ser lo suficientemente informativos y en idioma español para permitir la fácil comprensión del usuario y debe estar diseñado de forma que sea más fácil de utilizar para el usuario.

Hardware:

RNF3: El servidor deberá cubrir las siguientes características o contar con una variante equivalente a servidores web con memoria RAM: 4 GB o más, con posibilidades de expansión en caso de ser necesario. Capacidad de almacenamiento: 1TB.

Interfaz:

RNF4: El diseño de las interfaces que se proponen para el sistema a implementar está basado en las premisas de la organización y sencillez de la información a mostrar.

Seguridad:

RNF6: El sistema requiere la autenticación como primera acción, con un nombre de usuario único y una contraseña, que deben ser de conocimiento exclusivo de la persona que se autentica. Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso al sistema.

Rendimiento:

RNF7: Se debe garantizar que la respuesta a peticiones de los usuarios del sistema sea en un período de tiempo breve (15 de segundos), al igual que la velocidad de procesamiento de la información.

2.3 Planificación del *Sprint Backlog*:

Para la planificación del *sprint* se asignaron tareas por 15 días como máxima duración y se realizarán 2 *sprint* para un total de en un mes tener todos los requisitos implementados.

Tabla 3 Sprint 0

ID	Tareas	Estado	Tiempo Estimado	Tiempo Inicial	Tiempo final
T1	Instalación y configuración del entorno de desarrollo	Resuelta	72h	15/09/2022	18/09/2022
T2	Implementación de los requisitos funcionales 6,7,8,9,10	Resuelta	72h	18/09/2022	21/09/2022
T3	Implementación de los requisitos funcionales 11,12,13,14	Resuelta	72h	21/09/2022	24/09/2022
T4	Implementación de los requisitos funcionales 19,20,21,22	Resuelta	72h	24/09/2022	27/09/2022
T5	Implementación de los requisitos funcionales 23,24,25,26	Resuelta	72h	27/09/2022	30/09/2022

Tabla 4 Sprint 1

ID	Tareas	Estado	Tiempo Estimado	Tiempo Inicial	Tiempo final
T6	Implementación de los requisitos 15,16,17,18	Resuelta	168h	01/10/2022	07/10/2022
T7	Implementación de los requisitos funcionales 1,2,3,4,5	Resuelta	192h	08/10/2022	15/10/2022

2.4 Historias de Usuario

Las historias de usuario (HU) 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 de *software* desde el punto de vista del usuario, con el lenguaje que éste emplearía[CITATION Ale18 \l 3082]

A continuación, se muestra el nombre de cada una de las historias de usuario del sistema por cada requisito funcional:

HU1 Autenticar usuario

HU2 Registrar usuario

HU3 Editar usuario

HU4 Eliminar usuario

HU5 Visualizar lista de usuarios

HU6 Crear contrato

HU7 Editar contrato

HU8 Buscar contrato

HU9 Eliminar contrato

HU10 Visualizar lista de contratos

HU11 Crear ejecutor

HU12 Editar ejecutor

HU13 Visualizar lista de ejecutores

HU14 Eliminar ejecutor

- HU15** Crear dirección
- HU16** Editar dirección
- HU17** Visualizar lista de direcciones
- HU18** Eliminar dirección
- HU19** Crear empleado
- HU20** Editar empleado
- HU21** Visualizar lista de empleados
- HU22** Eliminar empleado
- HU23** Crear cliente
- HU24** Editar cliente
- HU25** Buscar cliente
- HU26** Eliminar cliente

Tabla 5 Historia de Usuario del requisito Crear contrato

Historia de Usuario	
Nombre: Crear contrato	
Número: 6	Usuario: Empleado y Ejecutor.
Prioridad: Alta	Riesgo en desarrollo: Alto
Programador: Xavierly Prats Ennis	Iteración asignada: 1
Tiempo estimado: 2h	Tiempo real: 6h
Descripción: El sistema debe permite insertar un contrato a la base de datos.	
Observaciones: Se completan los campos correspondientes del contrato: nombre del empleado, nombre del ejecutor, nombre del cliente, fecha de inicio del contrato, fecha de finalización del contrato, tipo de empresa, descripción del contrato, importe y horas de trabajo.	

Tabla 6 Historia de Usuario del requisito Editar empleado

Historia de Usuario	
Nombre: Editar empleado	
Número: 15	Usuario: Administrador
Prioridad: Alta	Riesgo en desarrollo: Alto
Programador: Xavierly Prats Ennis	Iteración asignada:1
Tiempo estimado:1h	Tiempo real:2h
Descripción: El sistema debe permitir editar los campos de un empleado registrado en la base de datos.	
Observaciones: Se modifican los campos deseados del empleado.	

Tabla 7 Historia de Usuario del requisito Autenticar usuario

2.5

Historia de Usuario	
Nombre: Autenticar usuario	
Número: 1	Usuario: empleado /ejecutor /administrador
Prioridad: Alta	Riesgo en desarrollo: Alto
Programador: Xavierly Prats Ennis	Iteración asignada:1
Tiempo estimado:1h	Tiempo real:2h
Descripción: El sistema debe permitir registrar un usuario.	
Observaciones: Para acceder al sistema el usuario debe con anterioridad crearse una cuenta en el sistema y se completan los campos correspondientes de la autenticación: usuario y contraseña.	

Arquitectura

La arquitectura de *software* es un conjunto de patrones que definen la organización de un sistema, sus componentes, el ambiente, y los principios que orientan su diseño y evolución. Los patrones arquitectónicos son la estructura del sistema que comprende a los elementos de *software*, las propiedades visibles externamente de dichos elementos y las relaciones entre ellos.[CITATION Jac13 \l 3082]

El principio más importante de la arquitectura Modelo-Vista-Controlador (MVC) es la separación del código en tres capas, dependiendo de su naturaleza. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador. La programación se puede simplificar si se utilizan otros patrones de diseño.[CITATION Rod19 \l 3082]

- ✓ Modelo: Es la representación de la información que maneja la aplicación. El modelo en sí son los datos puros que puestos en contexto del sistema que proveen de información al usuario o a la aplicación misma.
- ✓ Vista: Es la representación del modelo en forma gráfica, disponible para la interacción con el usuario.
- ✓ Controlador: Es la capa encargada de manejar y responder las solicitudes del usuario, procesa la información necesaria y modifica el Modelo en caso de ser necesario.

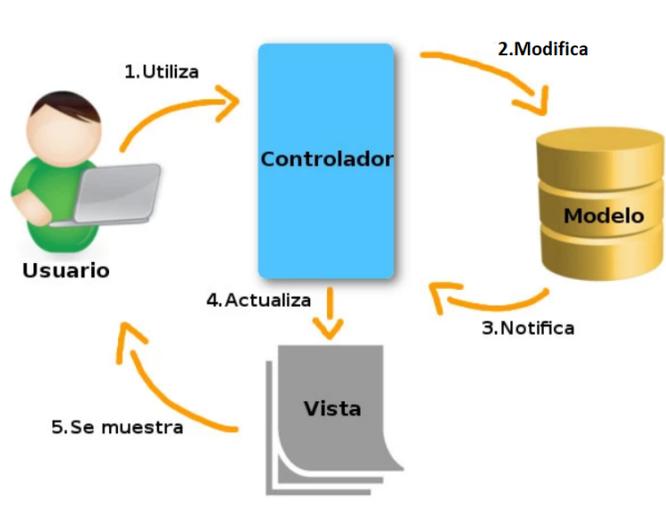


Figura 2 Arquitectura Modelo Vista Controlador (29)

A continuación, se muestra el diagrama de paquetes correspondiente a la arquitectura Modelo Vista Controlador de la presente investigación:

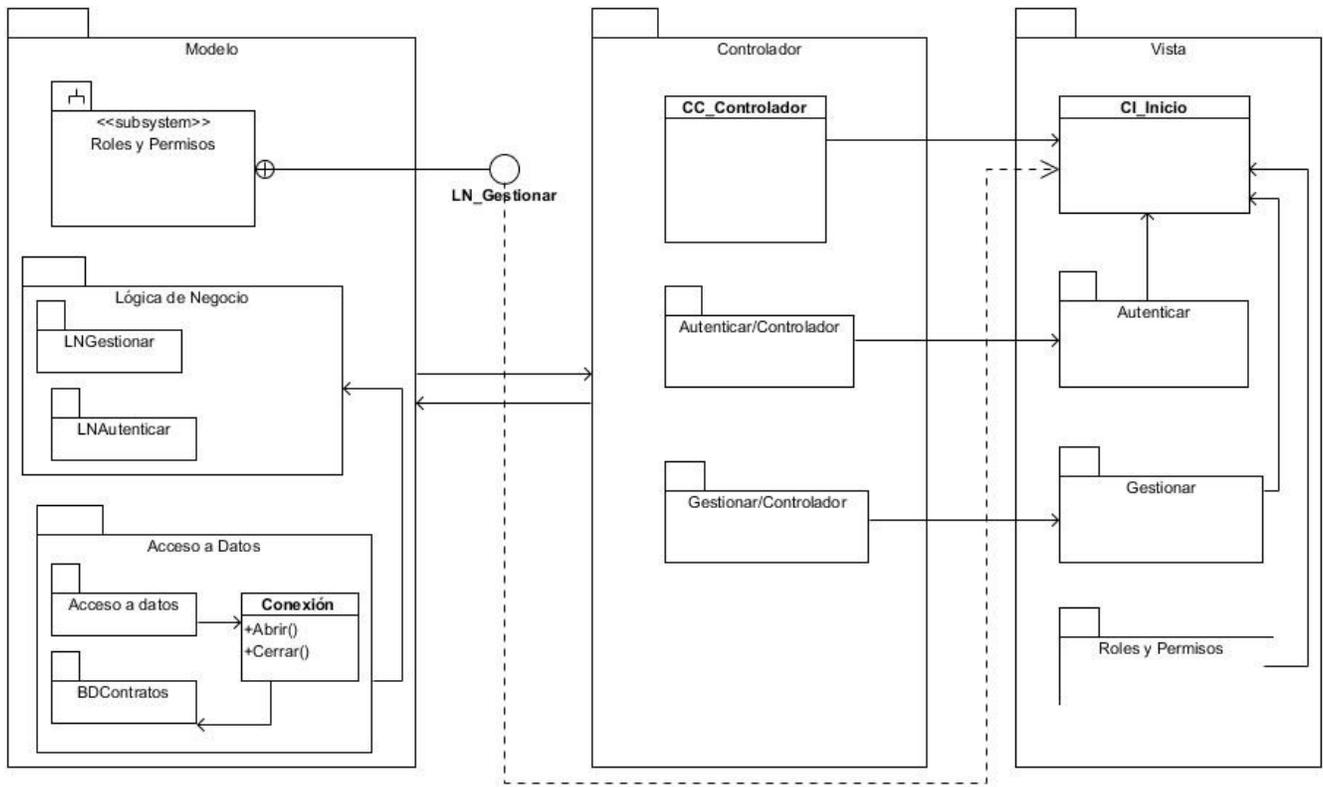


Figura 3 Diagrama de paquetes (Elaboración Propia)

Descripción del diagrama de paquetes:

- ✓ **Paquete Modelo:** En el modelo se agrupan los paquetes Lógica de Negocio, Acceso a datos y el subsistema Roles y Permisos. El paquete de Lógica de negocio se encarga de definir la forma en que se extraen y se analizan los datos para los paquetes gestionar y autenticar. El paquete de acceso a datos se encarga de la llamada y retorno de los datos que se obtienen de la base de datos contrato y la clase conexión es la que permite que se tenga acceso al uso de la base de datos. El paquete subsistema Roles y permisos representa una funcionalidad que será consumida desde el backend de la aplicación y que ya se encuentra implementada.
- ✓ **Paquete Controlador:** Él controlador se encarga de enviar las peticiones al

modelo que son recibidas desde la vista.

- ✓ **Paquete Vista:** En la vista se muestran cada una de las interfaces con las que interactúa el usuario.

2.6 Diagrama de clases de diseño

Un Diagrama de Clase del Diseño muestra la especificación de las clases de una aplicación, sus asociaciones, atributos y métodos, interfaces, navegabilidad y dependencias. El diagrama de clases recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura estática del sistema. [CITATION Pre10 \l 3082]

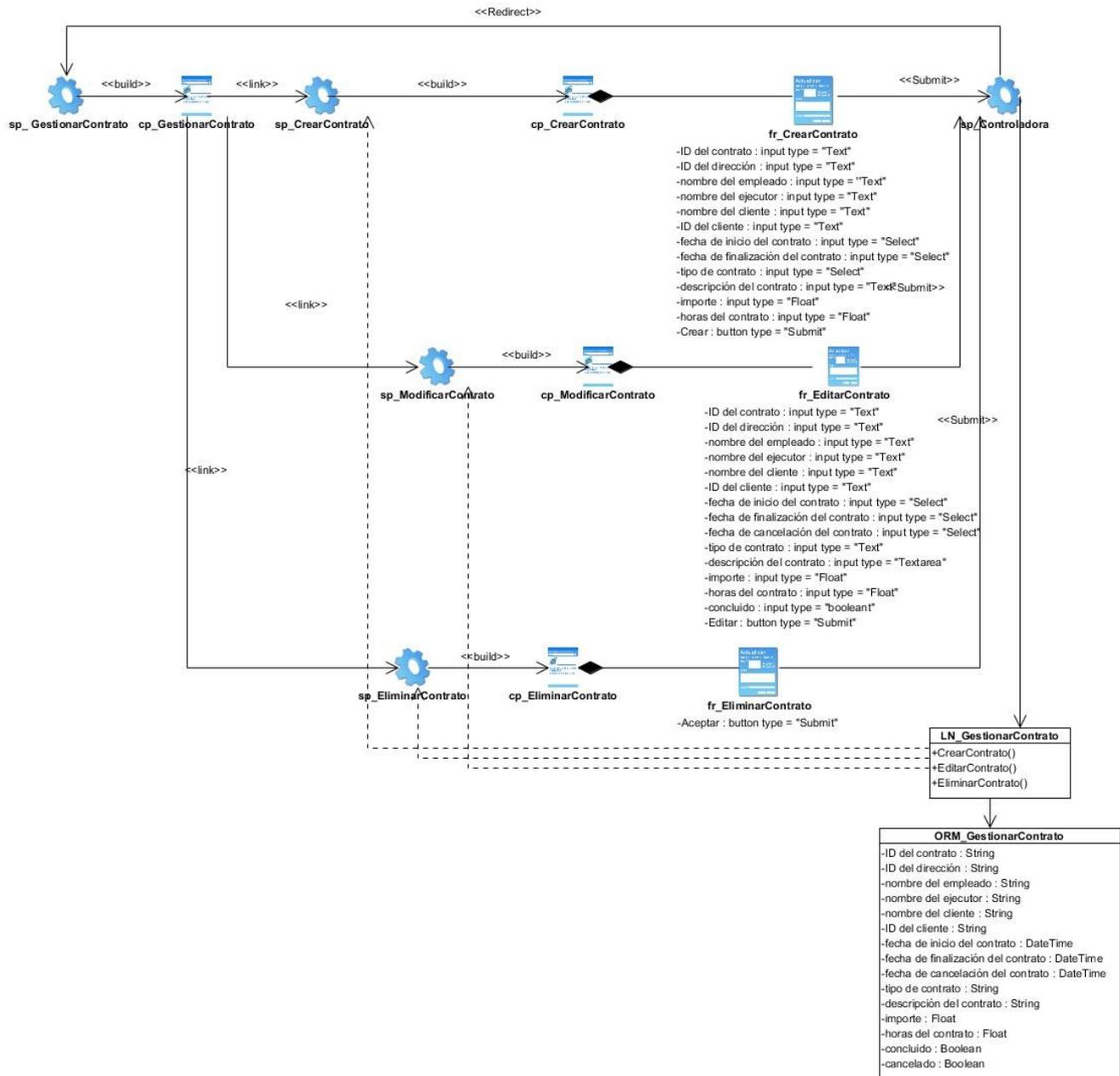


Figura 4 Diagrama de Clases del Diseño Gestionar Contrato.

Descripción de los elementos del Diagrama:

ServerPage sp: La clase página servidora es la encargada de construir la página cliente que es la que se muestra como interfaz.

ClientPage cp: La clase página cliente es la interfaz con la cual el usuario interactúa y envía las peticiones al sistema.

FormPage fr: La clase *Form* representa el formulario en el cual el usuario introduce los datos en la página cliente y mediante una relación *submit* envía los datos a la clase *server page* controladora.

2.7 Patrones de diseño

Los patrones GRASP se utilizan para la asignación de responsabilidades como sus siglas en inglés lo indican (*General Responsibility Assignment Software Patterns*), pero más que patrones se consideran una serie de buenas prácticas de aplicación recomendable en el diseño de software nos guían para ayudarnos a encontrar los patrones de diseño. [CITATION Cra05 \l 3082]

Controlador: Un Controlador es un objeto de interfaz no destinado al usuario, se encarga de manejar un evento del sistema a una clase que represente el sistema global. En el sitio, los eventos generados por el usuario son redirigidos a una clase controladora que realiza las operaciones solicitadas.



Figura 5 Patrón Controlador

Experto: Asigna la responsabilidad de hacer una labor a aquellos objetos que tienen la información para hacerlo. En el sistema, se pone de manifiesto el uso de este patrón, cuando se realizan cambios en los datos de algún contrato. La clase Contrato contiene los atributos necesarios para realizar las modificaciones, esta clase se puede llamar clase experta en información.



Figura 6 Patrón Experto

Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos para lograr menor dependencia y mayor oportunidad en la reutilización del código. Si se asignan bien el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización. Dentro del sistema este patrón se evidencia en la clase `Crear_Contratos` la cual es la encargada de tomar los nuevos datos que el usuario introduce para crear un contrato.

2.8 Modelo Entidad Relación

Un modelo de datos es un conjunto de herramientas conceptuales encargadas de describir la representación de los datos en una base de datos. Los mismos comprenden aspectos relacionados con estructuras y tipos de datos, operaciones y restricciones [CITATION Amp12 \l 3082]. A continuación, se expone el Modelo de datos, el cual fue generado para la presente investigación:

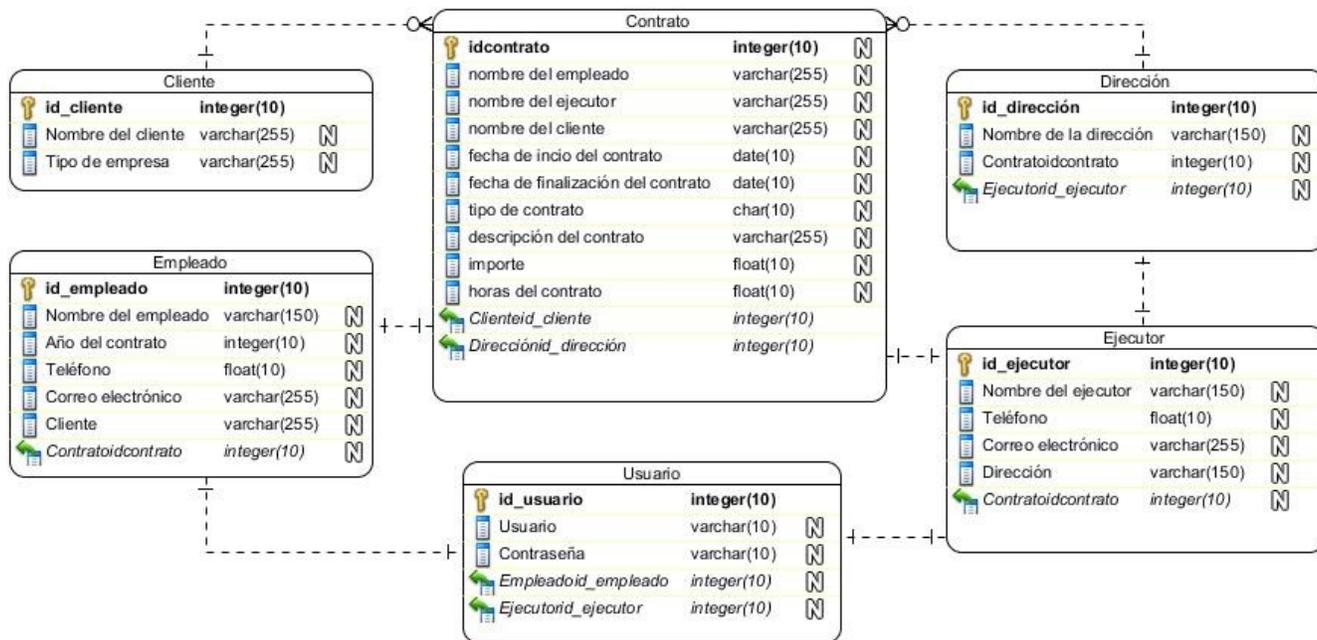


Figura 7 Modelo Entidad Relación de la Base de datos

2.9 Diagrama de Despliegue:

Un Modelo de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos [CITATION Joh \l 3082]. Un Nodo es un elemento de hardware o software.

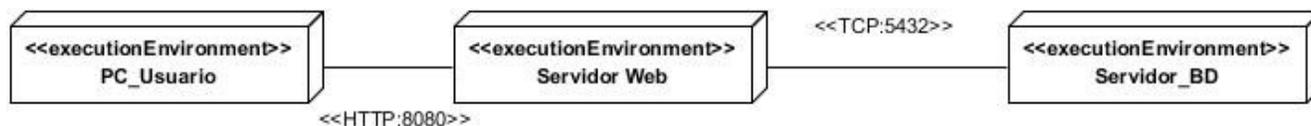


Figura 8 Diagrama de Despliegue

Descripción del Diagrama de Despliegue:

- ✓ **PC_Usuario:** Representa las computadoras clientes que se conectan al servidor de aplicaciones.
- ✓ **Servidor web:** Representa el servidor donde se encuentra instalada la aplicación web. Este accede al servidor de Base de Datos para el manejo de la información mediante el protocolo TCP/IP.
- ✓ **Servidor de Base de datos:** Es donde se almacena toda la información de la aplicación.
- ✓ **TCP:** Protocolo para conectar el servidor de aplicaciones con las bases de datos.
- ✓ **HTTP:** Protocolo de transferencia utilizado para conectar la computadora del cliente con el servidor donde está el sistema.

Conclusiones del capítulo

En este capítulo se evidencian las principales características y elementos significativos de la propuesta de solución, así como los diferentes requisitos funcionales y no funcionales.

1. La definición de los requisitos funcionales y no funcionales (*Product Backlog*) obtenidos a partir del proceso de identificación de los requisitos, garantizó que la solución responda a las necesidades del cliente, sirviendo de guía para el desarrollo de las funcionalidades del sistema de gestión de contrato.
2. Los artefactos generados constituyeron una guía fundamental para el desarrollo del sistema. La arquitectura seleccionada, (MVC) permitió definir la estructura del *software*.
3. El modelo de despliegue propuesto permitió exponer como se relacionan los elementos de *hardware* del sistema.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se abordarán los aspectos fundamentales del proceso de pruebas. Se define la estrategia de pruebas y se analizan los resultados de las pruebas realizadas que permiten evaluar la calidad de la propuesta de solución.

3.1 Pruebas de software

Las pruebas de *software* consisten en la dinámica de la verificación del comportamiento de un programa en un conjunto finito de casos de prueba, debidamente seleccionados de por lo general infinitas ejecuciones de dominio, contra la del comportamiento esperado. Son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa u ordenador; probando el comportamiento del mismo.[CITATION Ecu22 \l 3082]

La prueba de *software* es un elemento crítico para la garantía del correcto funcionamiento del *software*. Entre sus objetivos están[CITATION Ecu \l 3082]:

- ✓ Detectar defectos en el *software*.
- ✓ Verificar la integración adecuada de los componentes.
- ✓ Verificar que todos los requisitos se han implementado correctamente.
- ✓ Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el *software* al cliente.
- ✓ Diseñar casos de prueba que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

3.2 Estrategia de pruebas de software

Una estrategia de prueba de *software* proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuando se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requieren. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados. [CITATION Pre10 \l 3082] Una estrategia de prueba de *software* debe ser lo suficientemente flexible para promover un uso

personalizado de la prueba. Al mismo tiempo, debe ser suficientemente rígida para alentar la planificación razonable y el seguimiento de la gestión conforme avanza el proyecto.

Para la validación del sistema se propone realizar la siguiente estrategia de pruebas:

- ✓ **Pruebas funcionales:** Están basadas en ejecución, revisión y retroalimentación de las funcionalidades implementadas para el *software* y las especificaciones definidas por el usuario. Con estas pruebas se busca evaluar el sistema mediante modelos de pruebas. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro. [CITATION Pre10 \l 3082]

3.2.1 Aplicación de las pruebas

A continuación, se muestra la aplicación de las pruebas que se le efectuaron a la solución:

Pruebas Funcionales: En este tipo de pruebas se ejecutan los distintos servicios prestados con datos correctos e incorrectos. En caso de que los datos sean incorrectos se verifica que los mensajes de error sean los deseados y en el caso opuesto que los resultados sean los esperados. [CITATION Pre10 \l 3082]

Método de Prueba de Caja Negra:

El método de caja negra se aplica a la interfaz del *software*, examina algún aspecto funcional de un sistema que tiene poca relación con la estructura lógica interna del *software*. Se concentran en los requisitos funcionales del *software*, tratando de encontrar los siguientes errores: [CITATION Pre10 \l 3082]

- ✓ Funciones incorrectas o faltantes.
- ✓ Errores de interfaz.
- ✓ Errores de estructuras de datos o en acceso a bases de datos externas.
- ✓ Errores de comportamiento o desempeño.
- ✓ Errores de inicialización y término.

Los casos de pruebas pretenden demostrar que:

- ✓ Las funciones del *software* son operativas.
- ✓ La entrada se acepta de forma correcta.

- ✓ Se produce una salida correcta.
- ✓ La integridad de la información externa se mantiene.

Para el diseño de los casos de pruebas del método de Caja Negra se utilizó la Técnica de Partición Equivalente, la cual divide el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba. La partición equivalente se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse. [CITATION Pre10 \l 3082]

Tabla 8 Partición de equivalencia del caso de prueba "Crear contrato". (Elaboración propia)

Escenario	Descripción	Variable 1 (Campo Nombre del ejecutor)	Variable 2 (Importe)	Respuesta del sistema	Flujo central
1.1 Insertar datos correctos.	El usuario introduce los datos del contrato correctamente.	V	V	El sistema almacena la información introducida y se crea el contrato satisfactoriamente.	1-El usuario presiona en empleados crear un nuevo contrato 2-Se muestra un formulario con los campos a rellenar para crear el contrato. 3-Se presiona el botón Guardar.

<p>1.2 Insertar datos inválidos.</p>	<p>El usuario introduce datos erróneos.</p>	<p>V</p>	<p>I</p>	<p>El sistema muestra en la esquina superior derecho un mensaje con los campos incorrectos.</p>	<p>1-El usuario presiona en empleados crear un nuevo contrato 2-Se muestra un formulario con los campos a rellenar para crear el contrato. 3-Se presiona el botón Guardar. 4-El sistema muestra un mensaje con los campos erróneos.</p>
--------------------------------------	---	----------	----------	---	--

<p>1.3 Dejar campos vacíos.</p>	<p>El usuario deja campos vacíos.</p>	<p>I</p>	<p>I</p>	<p>El sistema muestra el mensaje "Campo inválido" junto con el campo que se encuentra vacío.</p>	<p>1-El usuario presiona en empleados crear un nuevo contrato 2-Se muestra un formulario con los campos a rellenar para crear el contrato. 3-Se presiona el botón Guardar. 4-El sistema muestra un mensaje con los campos erróneos en color rojo.</p>
---------------------------------	---------------------------------------	----------	----------	--	---

Resultados de las pruebas utilizando el método de caja negra.

Las pruebas funcionales aplicadas a la propuesta de solución son basadas en el diseño de casos de pruebas, donde se recogen los escenarios correspondientes a cada requisito funcional especificado. Para la realización de las pruebas se ejecuta cada caso de prueba utilizando datos válidos e inválidos en un total de 4 iteraciones. En la primera iteración se realizaron 9 casos de pruebas obteniendo 3 no satisfactorias, en una segunda iteración se realizaron 8 casos de prueba donde se obtuvieron 2 no satisfactorias, en una tercera iteración se realizaron 5 casos de pruebas de la cual se obtuvo 1 no satisfactoria y en una cuarta iteración se realizaron 4 casos de prueba donde se obtuvieron 2 no satisfacciones. Las no conformidades obtenidas en cada iteración fueron solucionadas al terminar las mismas, obteniendo un sistema que cumple de forma total con todas las funcionalidades requeridas por el cliente.

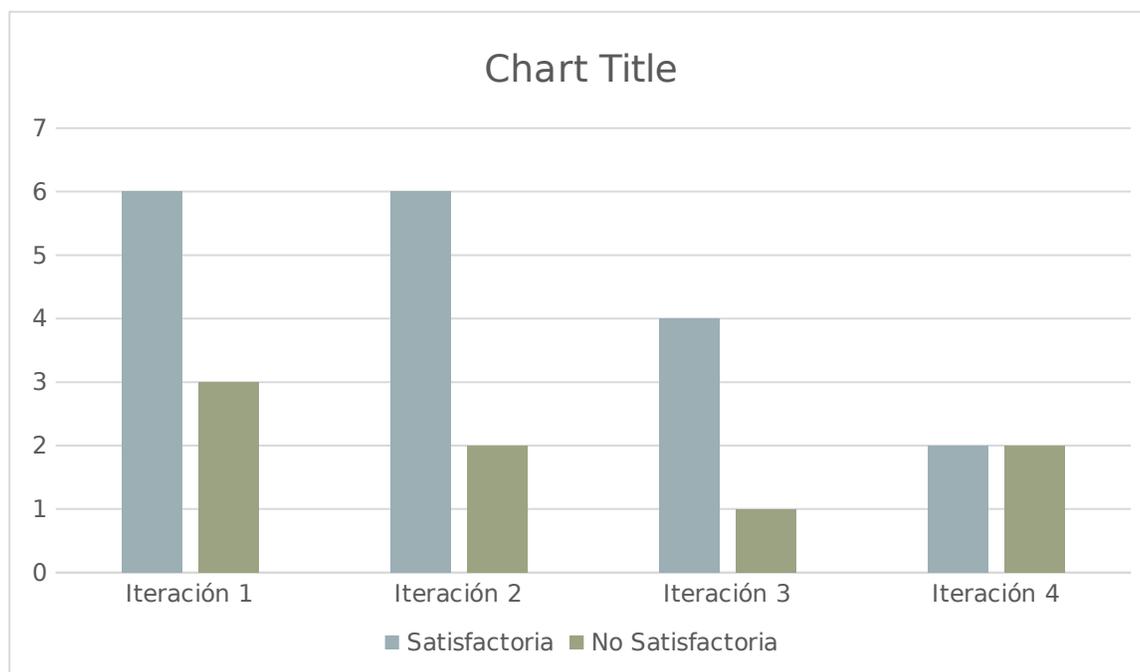


Figura 9 Resultados de la prueba de caja negra (Elaboración laboral)

Método de Prueba de Caja Blanca:

La prueba de caja blanca se basa en el examen cercano de los detalles de procedimiento sobre el código del software. Es considerada uno de los tipos de pruebas más importantes que se les aplican a los programas informáticos, logrando como resultado que disminuyan en un gran porcentaje el número de errores existentes en los sistemas y, por ende, se logre una mayor calidad y confiabilidad.

```
@api.onchange('company_id')
def onchange_company(self):
    if self.origin:
        return { "warning": {
            'title': _("Warning"),
            'message': _("To avoid multi company issues (loosing the access to your
previous))
        }}
    }
```

Figura 10 Método onchange_company_id

Tabla 9 Camino básico

Fórmula 1	Fórmula 2	Fórmula 3	Grafo
$V(G)=(a-n) +2$ $= (5-5) +2$ $=2$	$V(G)=p+1$ $=1+1$ $=2$	$V(G)=r$ $=2$	
<p>Complejidad ciclomática</p> <p>Como se puede observar, después de aplicadas las formulas 1, 2 y 3 a la funcionalidad onchange_company_id posee una complejidad ciclomática igual a 2, lo cual demuestra que el grafo está bien diseñado y que existen 2 caminos lógicos e independientes.</p>		<p>Prueba de camino básico</p> <p>Una vez calculada la complejidad ciclomática se define como límite superior 2, lo que indica que hay que realizarle al código 2 pruebas, para garantizar que este se ejecute completamente al menos una vez.</p> <p>El total de caminos independientes fue de 2 que a continuación se muestran:</p> <ul style="list-style-type: none"> ✓ 1,2,3,4 y 5 ✓ 1,2,4 y 5 	

Descripción de los elementos de la tabla:

A la cantidad de aristas del grafo.

N la cantidad de nodos del grafo.

P el número de nodos predicados contenidos en el grafo.

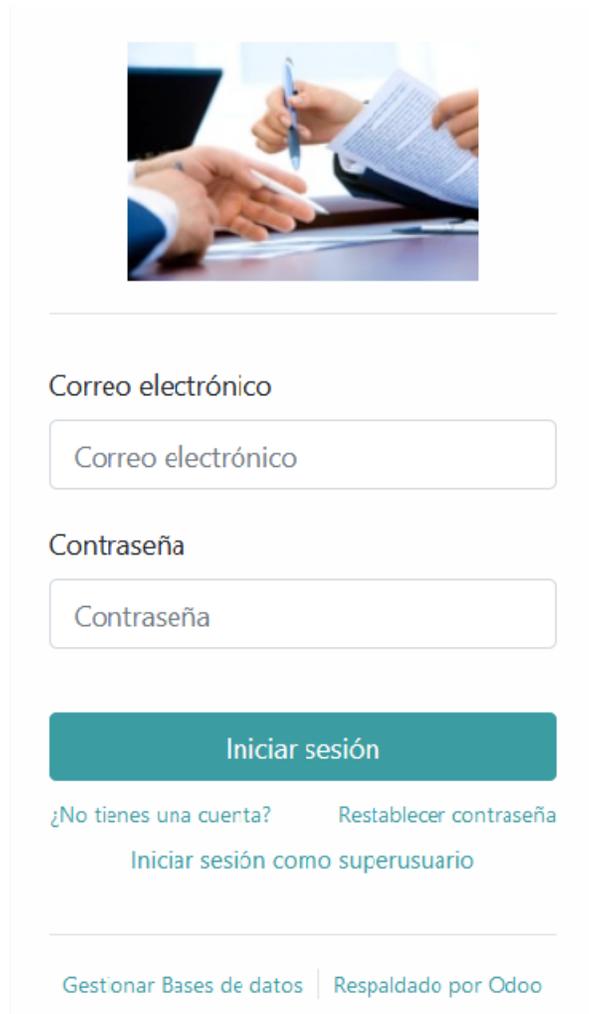
R representa la cantidad de regiones en el grafo.

Tabla 10 Descripción del camino básico

Descripción	Funcionamiento
Condición de ejecución	@api.onchange("")
Datos de entrada	1,2,3,4,5
Resultados	Se puede recorrer todo el grafo.
Resultados de prueba	Satisfactorios

Interfaz principal del sistema de gestión de contratos:

Una vez desarrollado el sistema de gestión de contratos, es posible visualizar las pantallas principales del mismo donde se observa el resultado obtenido durante la implementación de los requisitos funcionales descritos en el capítulo 2.



Correo electrónico

Contraseña

[Iniciar sesión](#)

[¿No tienes una cuenta?](#) [Restablecer contraseña](#)

[Iniciar sesión como superusuario](#)

[Gestionar Bases de datos](#) | [Respaldado por Odoo](#)

Figura 11 Captura de pantalla del sistema (Autenticar usuario)

Referencia de contrato

Nombre del empleado	Klaudia Perez García 
Nombre del ejecutor	
Nombre del cliente	
Fecha de inicio del contrato	23/11/2022
Fecha de finalización de contrato	
Tipo de empresa	
Descripción del contrato	
Departamento	
Importe	0,00
Horas del contrato	0,00

Figura 12 Captura de pantalla del sistema (Crear contrato)

Conclusiones del capítulo

En este capítulo se realizó la estrategia de pruebas para que fuera más fácil el entendimiento del código del programador y facilitar el mantenimiento futuro del sistema. Se cumplió con el objetivo planteado en el diseño teórico, se llevó a cabo la realización de un adecuado proceso de pruebas durante el desarrollo de la aplicación, lo cual permitió alcanzar un *software* de elevada calidad y comprobada satisfacción de los requisitos convenidos. Luego de cada cambio significativo en el sistema, se realizaron pruebas de caja negra con la técnica de partición de equivalencia y pruebas de caja blanca usando la técnica de camino básico.

CONCLUSIONES FINALES

Al completar el desarrollo de la solución informática se le dio cumplimiento al objetivo planteado inicialmente, obteniendo como resultado principal un sistema de gestión de contratos para el Centro Internacional de La Habana.

- ✓ La definición de las herramientas, metodologías y tecnologías empleadas en el desarrollo del sistema de gestión de contratos, propició la identificación del ambiente de desarrollo de acuerdo a las necesidades del proyecto, lo que permitió que su implementación se realizara exitosamente.
- ✓ El sistema garantiza que se comporte de manera fiel y responda a las necesidades reales del Centro Internacional de La Habana mejorando los procesos de la gestión de contratos según las direcciones, asegurando la trazabilidad y brindando una mejor clasificación.
- ✓ La solución fue validada a partir de la definición correcta de una estrategia de pruebas que permitió comprobar el correcto funcionamiento del sistema de gestión de contratos para el Centro Internacional de La Habana, a partir de los requerimientos definidos por el cliente.

RECOMENDACIONES

Una vez culminado el sistema de gestión de contratos para el Centro Internacional de La Habana, se pudo comprobar que los objetivos trazados al comenzar la investigación fueron resueltos de manera satisfactoria, aunque se debe tener en cuenta que esta es la primera versión del sistema, por tanto, se recomienda para futuras versiones:

- ✓ Implementar una funcionalidad que se integre con el sistema existente y permita las notificaciones mediante correo electrónico.

REFERENCIAS

1. Miranda Roque. Gestipolis. Obtenido de La Gestión Comercial y la Venta. [En línea] 17 de febrero de 2018. [Citado el: 15 de septiembre de 2022.] <https://www.gestipolis.com/la-gestion-comercial-la-venta/>.
2. CIH S.A. [En línea] [Citado el: 15 de septiembre de 2022.] <https://www.cihservicios.com/es/about-us/>.
3. Rodriguez, Lianet Cabrera. *Sistema para la Gestión de contratos basado en indicadores, orientado a la Gestión de proyectos*. Carretera San Antonio : s.n., 2017.
4. Maria, Jorge Lopez Santa. *Los Contratos*. Segunda edición actualizada. Chile : Editorial jurídica de Chile, 1998. 1.
5. Camacho, César Abraham Burgos. *Aplicación web para la gestión de contratos en el área de logística de visualtec creativo*. 2018.
6. Luna, Kalu. *Facultad de administración de empresas y negocios internacionales Tema: sistemas de gestión para las empresas comerciales*. 2014.
7. Naranjo, Francisco José. El Blog de Seidor. [En línea] 15 de enero de 2015. [Citado el: 29 de octubre de 2022.]
8. Cristaldo, Patricia y Kloster, Miriam. *Sistemas de Gestión*. 2016.
9. Desoft. [En línea] [Citado el: 15 de septiembre de 2022.] <https://www.desoft.cu/es/products/332>.
10. Juro. [En línea] 2009. www.juro.com.
11. Volody Contract LifeCycle Management. [En línea] 2017. [Citado el: 15 de septiembre de 2022.] <https://volodycontractlifecyclemangement.com/volody>.
12. Navarro Cadavid, Andrés, Fernández Martínez, Juan Daniel y Morales Vélez, Jonathan . Revisión de metodologías ágiles para el desarrollo de software Prospectiva. [aut. libro] Universidad automática del Caribe. 2013, Vol. 11, 2, págs. 30-39.
13. Gauchat, Juan Diego. *El gran libro de HTML5, CSS3 y Javascript*. Marcombo : s.n., 2012.
14. Mestras, Juan Pavón. *Bootstrap, aplicaciones web/ sistemas web*. Dep. Ingeniería del Software e Inteligencia Artificial Facultad de Informática Universidad Complutense Madrid. Madrid : s.n., 2011.
15. Luna, Fernando. *JavaScript-Aprende a programar en el lenguaje de la web*. s.l. : RedUsers, 2019.
16. Walker, Jonathan S. *Python: La guía definitiva para principiantes para dominar Python* . s.l. : Babelcube Inc, 2018.
17. Fontela, Carlos. *UML: modelado de software para profesionales*. s.l. : Alpha Editorial, 2012.
18. Rahwanto, Henderi Dr Untung Rahardia Efana. *UML Powered Design System Using Visual Paradigm*. s.l. : Literasi Nusantara Abadi, 2022.
19. Holovaty, Adrian y Kaplan-Moss, Jacob. *The Book of Django*. 2008.
20. Rubio, Daniel. *REST services with Django En Beginning Django*. s.l. : Apress Berkeley, 2017. págs. 549-566.
21. Castillo, Alberto Ayoze. *Web Programming Course: JavaScript, Ajax y jQuery*. s.l. : Campus Academy, 2017.
22. Bachman. *The programer as navigator*. s.l. : Communications of the ACM, 2015.
23. Niño, Jesús. *Servidores de aplicación web*. s.l. : Editex, 2011.
24. Arman, Craig. *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development*. India : Pearson education, 2005.
25. Somerville, Ian. *Ingeniería de software*. México : Pearson education Inc, 2011.
26. —. *Ingeniería de Software*. 9na. México : Pearson Education Inc., 2011.
27. Menzinsky, Alexander. *Historias de usuario*. Ingeniería de requisitos ágil. 2018.
28. Kaplan-Moss, Jacob. *El libro de Django*. 2013.
29. Rodríguez, Mayté y Mon, Manuel Alejandro. *Sistema basado en web para la selección y asignación de la práctica profesional*. 2019.

30. Pressman, Roger. *Ingeniería del software un enfoque práctico*. 2010. pág. 767.
31. Gaona, Amparo López. *En Ciencia, Posgrado. El modelo Entidad-Relación* . 2012.
32. Sarmiento, Johana. *UML Diagrama de despliegue. Visión general del diagrama de despliegue*.
33. EcuRed. [En línea] [Citado el: 30 de octubre de 2022.] https://www.ecured.cu/Pruebas_de_softwa-re.
34. EcuRed. [En línea] www.ecured.cu.
35. Peño, José Manuel Sanchez. *Pruebas de software.Fundamentos y técnicas*. 2015.
36. Torres, Amalia. *Pruebas de software*. Chile : Nueva Era, 2016.

ANEXOS

Entrevista a un empleado del Centro Internacional de La Habana.

1- ¿Cómo se realiza la Gestión de contratos en el CIH?

2- ¿Con qué herramienta se realiza la gestión de contratos en el CIH?

3- ¿La herramienta con la que realizan la contratación les permite tener información inmediata de los contratos?

Técnica partición de equivalencia del caso de prueba Autenticar usuario:

Escenario	Descripción	Variable 1 (Usuario)	Variable 2 (Contraseña)	Respuesta del sistema	Flujo Central
1.1 Insertar datos correctos.	El usuario introduce sus datos correctamente.	V	V	El sistema permite que el usuario acceda al sistema y le muestra en pantalla la página de inicio.	1-Se introducen los datos en los campos de texto. 2-Se selecciona el botón iniciar sesión.

1.2 Insertar datos inválidos.	El usuario introduce datos erróneos.	I	V	El sistema muestra un mensaje al usuario un mensaje con los campos incorrectos.	1-Se introducen los datos en los campos de texto. 2-Se selecciona el botón iniciar sesión. 3-El sistema muestra un mensaje con el error.
1.3 Dejar campos vacíos.	El usuario deja campos vacíos.	I	I	El sistema muestra los campos vacíos de color rojo.	1-Se introducen los datos en los campos de texto. 2-Se selecciona el botón iniciar sesión. 3-El sistema muestra los campos vacíos de

					color rojo.
--	--	--	--	--	-------------