

UNIVERSIDAD DE MATANZAS
FACULTAD DE CIENCIAS TÉCNICAS
DEPARTAMENTO DE INFORMÁTICA



**Módulo para la gestión de contratos y facturas de la
empresa Aicros.**

Trabajo de diploma en opción al título de Ingeniero Informático

Autor: Roxana Naranjo Benítez

Tutora: M. Sc. Mayli Estopiñán Lantigua

Matanzas, 2022

“Para mí la programación es más que un importante arte práctico. También es un desafío gigantesco en los fundamentos del conocimiento.”

Grace Hope

A mi tía Beba, una gran ingeniera

Agradecimientos:

Principalmente a mis padres por estar conmigo apoyándome, en cada paso del camino.

A mi tío que me enseñó a reír en los momentos más difíciles.

A mi hermano, que fue la primera persona que me mostró lo genial que es la programación.

A todos mis abuelos, en especial a aquellos que ya no están.

A mis compañeros de carrera, fue genial conocerlos a todos.

A David, ¡mira que te molestaba con dudas!

Al colectivo de profesores de la Facultad de Informática de la Universidad de Matanzas "Camilo Cienfuegos" le agradezco su entrega en todos estos años que contribuyó significativamente a mi formación profesional.

¡Gracias a todos!

Declaración de autoría

Yo, Roxana Naranjo Benítez, me declaro única autora del Trabajo de Diploma de título: Módulo para la gestión de contratos y prefacturas de la empresa Aicros, además autorizo a la entidad donde se realizó la investigación y a la Universidad de Matanzas sede Camilo Cienfuegos para hacer uso de este con el fin que estime más conveniente.

Firma del autor

Opinión del tutor

Resumen

A través de plataformas digitales los cuentapropistas pueden obtener de otros actores similares todo tipo de bienes y servicios de forma rápida, desde cualquier país del mundo y con una importante reducción de costes de transacción y de búsqueda. En la empresa Aicros el proceso de contratación y facturación se realiza de forma manual y posteriormente dichos datos son entrados a diversas aplicaciones, haciendo de esta una tarea engorrosa y repetitiva. Se trazó como objetivo general desarrollar un módulo que incremente la eficiencia del proceso de gestión de contratos y prefacturas para la empresa Aicros. Para alcanzar el objetivo propuesto, las tecnologías que se utilizaron fueron: *Nest.js* para el *backend*, *Angular* en el *frontend*, *PostgreSQL* para la base de datos, como metodología XP y las herramientas seleccionadas fueron *VisualCode*, *VisualParadigm* y *PgAdmin*.

Summary

Trough digital platforms, the self employed can obtain all kinds of goods and services from other similar actors, quickly, from any country in the world and with a significant reduction in transaction and search costs. In the Aicros company, the contracting and billing process is carried out manually and later said data is entered various applications making this a cumbersome and repetitive task. The general objective was to develop a module that increases the efficiency of the contract and pre-invoice management process for the company Aicros. To achieve the proposed objective, the technologies that were used were *Nest.js* for the backend, *Angular* for the frontend, *PostgreSQL* for the database, the XP methodology and the selected tools were *VisualCode*, *VisualParadigm* and *PgAdmin*.

Índice

Introducción.....	1
Gestión de Contratos	1
Gestión de facturas	2
Capítulo 1 Fundamentación teórica.....	6
Introducción.....	6
Descripción de los procesos que serán automatizados	7
1.2 Análisis crítico de la ejecución de los procesos de contratación y prefacturación	8
1.2. Análisis comparativo de propuestas existentes con la propuesta del trabajo.....	9
1.2.1 Sistema para la gestión de Oportunidades de Negocio para el Centro de Informática Industrial	9
1.2.2. Myros	10
1.2.3. Aportes de los sistemas homólogos estudiados.	10
1.3 Marco legal para la contratación en Cuba.	10
1.4 Metodología y herramientas seleccionadas en la implementación de la solución.	11
1.4.1 Marcos de trabajo (<i>Frameworks</i>).....	11
1.4.2 Lenguajes soportados por el <i>framework</i>	14
1.4.2.1 HTML5	14
1.4.2.2 CSS3	14
1.4.2.3 <i>TypeScript</i>	15
1.4.3 Entorno de desarrollo integrado (IDE):	15
1.4.4 <i>Visual Paradigm</i>	15
1.4.5 Gestores de Bases de datos	17
1.4.4 Sistema de control de versiones <i>Git</i>	19
1.4.6 Metodología de desarrollo	20
Arquitectura N capas	21
Conclusiones.....	25
Capítulo 2 Descripción de la solución propuesta.....	26
Introducción	26
2.1 Descripción de la solución	26
2.2 Etapa de planificación	27
2.2.1 Equipo de trabajo y roles.....	28
2.2.2 Historias de Usuario Iniciales	28
2.2.3 Plan de Iteraciones	29
2.3 Planificación del proyecto.....	30

2.5 Etapa de diseño.....	35
2.5.1 Prototipo de la interfaz de usuario.....	35
2.5.2 Tareas de Ingeniería	40
Conclusiones parciales del capítulo.....	45
Capítulo III, Elementos de prueba y Resultados obtenidos	46
Introducción.....	46
3.1 Elementos de prueba al <i>software</i>	46
3.2 Plan de pruebas	46
3.2 Análisis de los resultados obtenidos.	60
Conclusiones	60
Conclusiones	61
Recomendaciones	62
Bibliografía.....	63

Introducción.

La digitalización de la economía, fruto de los importantes avances tecnológicos de los últimos años, ha facilitado que emerja un nuevo modelo de mercado que está teniendo un crecimiento progresivo y constante, que queda englobado en lo que se ha denominado la economía colaborativa (*sharing economy*) o también denominada economía bajo demanda o consumo colaborativo: un conjunto heterogéneo y rápidamente cambiante de modos de producción y consumo por el que los agentes comparten, de forma innovadora, activos, bienes o servicios infrautilizados, a cambio o no de un valor monetario, valiéndose para ello de plataformas sociales digitales y, en particular, de internet (Comisión Nacional de los mercados y la competencia (CNMC), 2016). Su impacto y desarrollo está siendo extraordinario a nivel mundial, y parece que se cumple la conclusión que consideraba que la economía colaborativa era una de las 10 ideas que cambiarían el mundo (Walsh, 2011).

A través de plataformas digitales los particulares pueden obtener de otro particular todo tipo de bienes y servicios de forma rápida, desde cualquier país del mundo y con una importante reducción de costes de transacción y de búsqueda, lográndose una monetización de recursos infrautilizados y reduciéndose las asimetrías informativas y el riesgo de selección adversa (Cuenca, 2020).

Gestión de Contratos

La contratación masiva es el eje central de un proceso de cambios que trasciende debido a la dinámica de la economía actual. Las sociedades contemporáneas, incluyendo a Cuba, han sufrido ingeniosas transformaciones que han provocado una seria erosión en los principios conformadores del contrato, al punto que para muchos autores la institución está en crisis. Esto ha traído consigo nuevas formas de formación del contrato dígame una sustitución del modelo de contrato por negociación clásico por el modelo de contrato por adhesión, la aparición de la contratación electrónica, que no constituye un nuevo contrato sino un modo de generación del contrato que responde a las nuevas exigencias del tráfico debido al avance tecnológico (Urbon López, 2022).

La Contratación pública, es uno de los factores que permiten que un país pueda desarrollarse mediante la consecución de los objetivos y planes que un estado propone para garantizar la igualdad de oportunidades a pequeños sectores como las MiPymes. (Varela Vaca, 2018)

Cuando dos compañías desean hacer negocios entre ellas, un contrato especifica las actividades realizadas por ambas organizaciones y los términos a través de los cuales cada uno cumplirá con sus partes del acuerdo. Los contratos afectan la rentabilidad del negocio de una manera muy

grande debido al énfasis en los ingresos y los gastos. Cuando un contrato tiene una redacción deficiente, una organización puede perder incontables miles de dólares por un simple tecnicismo que no pueden identificar. La administración eficaz de contratos puede, en última instancia, crear una relación comercial poderosa y allanar el camino hacia una mayor rentabilidad a largo plazo, pero solo si se gestiona correctamente.

No es suficiente que una organización tenga profesionales para gestionar los contratos. El rendimiento de los trabajadores debe mejorar con la presencia de procesos y *softwares* para la gestión de contratos para así satisfacer las crecientes necesidades analíticas y de cumplimiento de la organización. Cuando se implementa con éxito una estrategia de gestión de contratos, las organizaciones pueden esperar ver que:

- Los beneficios comerciales esperados y los rendimientos financieros se están realizando
- El proveedor es cooperativo y receptivo a las necesidades de la organización
- No habrá conflictos contractuales
- La entrega de servicios es satisfactoria para ambas partes

La gestión de contratos requiere un nivel de flexibilidad para las partes involucradas y una voluntad de adaptar los términos del contrato para reflejar cualquier cambio en las circunstancias. Los problemas son inevitables, lo que significa que las organizaciones deben estar preparadas para lo inesperado y poder ajustar los términos del contrato cuando sea necesario.

La práctica tradicional en la administración de contratos muestra ineficiencia que solo pueden restarle eficacia a la organización. De ahí que, la integración de un servicio automatizado ayudará a reducir las horas de trabajo y automatizar los procesos asociados a la gestión de contratos, creando así más valor para una empresa (Gestión de contratos, 2022).

Gestión de facturas

Agilización de procesos, ahorro de dinero, horas de trabajo, de papel y espacio. La gestión digital de facturas supone muchas ventajas para las empresas, incluido el departamento de Recursos Humanos.

Hay estudios que indican que el ahorro de tiempo para una empresa puede superar las 250000 horas anuales (EGA Futura, 2022). Eso se traduce en dinero. Pero también implica ganar en operatividad y eficiencia, puesto que el equipo de finanzas y administración puede destinar ese tiempo ganado en realizar otras tareas. De continuar con los procesos manuales, es más probable que se necesiten más empleados para desempeñar el mismo volumen de trabajo.

Por cuestiones tan prácticas es por lo que la gestión automatizada de facturas de proveedores se ha convertido en una de las claves de la transformación digital empresarial.

Eficacia para las empresas de la gestión digital de facturas

Las empresas no pueden permitirse parar y para ello necesitan herramientas que ofrezcan la posibilidad de continuar con su actividad desde cualquier ubicación. Además de las ventajas antes mencionadas esta gestión automatizada puede suponer una gran oportunidad para los departamentos de Contabilidad y Recursos Humanos. Entre algunos ejemplos:

- Optimización de recursos.
- En caso de someter a la empresa a una auditoría el proceso será más sencillo.
- Se recibe información en tiempo real de las cuentas de cada área.
- La toma de decisiones se agiliza.
- Al eliminar los procesos manuales se reducen los errores.
- Como se consume menos papel se reduce la huella de carbono.
- Los archivos están más organizados. No se volverán a traspapelar y se encontrarán los documentos en cuestión de segundos.
- Se pueden realizar tareas de forma simultánea.
- Siempre es más fácil atraer talento cuando una empresa invierte en innovación tecnológica. Pero el área que responde a Recursos Humanos siempre es la gran ignorada. Si este proceso está informatizado se posee una gran diferencia con la competencia.
- Se reduce el tiempo de espera para pagos y cobros.
- Aumenta la seguridad ya que no se corre el riesgo de perderse durante el proceso de envío.
- Como los trabajadores ganan eficacia, aumenta su bienestar.
- Se puede integrar la herramienta de gestión automatizada con la de gestión de compras y pedidos.
- Es importante destacar que para gestionar o supervisar este proceso ya no es necesario estar en la oficina. Puede hacerse desde cualquier dispositivo móvil (Calvo, 2021).

La Empresa de Informática y Automatización para la Construcción (AICROS) perteneciente al Grupo Empresarial de Diseño e Ingeniería de la Construcción del Ministerio de la Construcción es la encargada de elaborar y comercializar sistemas y aplicaciones informáticas y soluciones integrales de informatización y automatización industrial, además de comercializar medios técnicos de computación y periféricos especializados, sus partes, piezas y accesorios, ofrece servicios telemáticos, utilizando la infraestructura de los suministradores públicos autorizados.

La gestión de los documentos, como la realización del procesamiento, tanto de los contratos como de las prefacturas y facturas, se hacen a partir de proformas que contienen los requisitos para la realización de estos, hechos en pdf, las cuales luego se imprimen y son llenadas con sus datos a mano. Los clientes tienen que acudir, a la entidad, más de una vez hasta que el proceso se culmine. Una vez terminado el proceso la información es almacenada en aplicaciones independientes de la tienda, quedando descentralizada. Esto trae como consecuencia un esfuerzo y un agotamiento mayor también al realizar el trabajo.

A partir de esta situación problemática se define como **problema de la investigación**: ¿Cómo contribuir a la eficiencia del proceso de contratación y facturación en la empresa Aicros?, teniendo como **objeto de estudio** el proceso de venta y contratación de los servicios de la empresa AICROS y siendo el **campo de acción** la informatización del proceso de ventas y contratación de servicios en la empresa AICROS.

Del problema anterior se deduce la siguiente **hipótesis**: Si se realiza un módulo para la gestión de contratos y facturas en la empresa Aicros, entonces se contribuirá a la eficiencia del proceso de contratación y facturación, con lo que se logra a su vez un fácil manejo para los prestadores de esta entidad.

Con el propósito de validar la hipótesis anterior se define como **objetivo general**: Desarrollar un módulo que contribuya a la eficiencia del proceso de gestión de contratos y facturas para la empresa Aicros.

Para lograr el cumplimiento del objetivo general se declaran los siguientes **objetivos específicos**:

1. Determinar el Marco Teórico sobre el proceso de gestión de contratos y facturas en la empresa Aicros.
2. Seleccionar las tecnologías y herramientas para el proceso de gestión de contratos y facturas en la empresa Aicros.
3. Diseñar un módulo para incrementar la eficiencia de la contratación y facturación, en línea, en la empresa Aicros.
4. Implementar el módulo para el proceso de gestión de contratos y facturas en la empresa Aicros.
5. Validar el módulo para el proceso de gestión de contratos y facturas en la empresa Aicros.

Método de investigación y técnicas empleadas.

Los métodos teóricos de análisis y síntesis, y el inductivo-deductivo fueron utilizados durante toda la investigación con el fin de analizar la funcionalidad y la existencia de *softwares* vinculados al proceso de contratación y facturación en línea, así como determinar la influencia de otros factores al problema. Nos permitió el empleo de diversas informaciones, recogidas en las fuentes bibliográficas que se consultaron, y la elaboración de las conclusiones y recomendaciones.

Como métodos empíricos se realizaron entrevistas a diferentes funcionarios de los departamentos Comercial y Jurídico, mediante los cuales se pudo efectuar una exploración preliminar del proceso, así como una apreciación de su desarrollo: cómo se registra, organiza, actualiza, crea y almacena actualmente la información, determinando los requisitos funcionales y no funcionales.

Para el desarrollo de la aplicación se utilizó la metodología *Extreme Programming (XP)*, la cual se basa en el seguimiento constante y la realización de diversas pruebas y pequeños ajustes en proyectos que necesitan agilidad o que están en constante cambio.

Se utiliza el método de estimación por Puntos de Función basado en ecuaciones matemáticas que permiten calcular el esfuerzo a partir de ciertas métricas de tamaño estimado.

Este trabajo de investigación consta de una Introducción en la que se trata el problema a investigar, los antecedentes, los objetivos, y los métodos de investigación. El cuerpo del trabajo está dividido en capítulos:

Capítulo I, Fundamentación teórica. Se trata del estado actual del negocio, el Marco Legal sobre el que se sostiene la contratación en línea, así como las herramientas de programación y la metodología de desarrollo seleccionadas.

Capítulo II, Descripción de la solución propuesta. Se expone la propuesta de solución al problema de investigación, la organización y disposición del proyecto mediante la presentación de una planificación inicial haciendo uso del marco de trabajo de desarrollo de software XP. Es analizado y estudiado los beneficios obtenidos mediante la implementación del proyecto de software.

Capítulo III, Elementos de prueba y Resultados obtenidos. Muestra las diversas pruebas realizadas al software para determinar la funcionalidad y calidad de este con el objetivo de entregar al cliente un producto terminado que satisfaga sus requerimientos.

El cuerpo del trabajo consta además de **Conclusiones, Recomendaciones, Bibliografía y Anexos.**

Capítulo 1 Fundamentación teórica

Introducción.

En este capítulo se abordan los conceptos y premisas fundamentales relacionadas con el proceso de contratación y facturación, que permitirán un mejor entendimiento de la solución planteada posteriormente y servirá como preámbulo para el desarrollo posterior de la investigación. En él se exponen los estudios previos realizados sobre el tema y se analizan aplicaciones similares a la que se pretende desarrollar. Se estará analizando además el Marco Legal bajo el que se rige la contratación en nuestro país y las metodologías y herramientas utilizadas.

Facturación Digital en América Latina

La factura electrónica se ha ido imponiendo en América Latina a lo largo de los últimos años, adquiriendo una relevancia significativa a lo largo de 2021. Los pioneros en su uso son Chile, México y Brasil, donde se ha masificado y avanza en procesos de cumplimiento fiscal, como la contabilidad electrónica para empresas con facturación de hasta 5 millones de pesos. El objetivo es facilitar estas tareas a los contribuyentes reduciendo costes.

Se implanta por primera vez en Chile, en 2003, para un grupo reducido de contribuyentes, convirtiéndose en obligatoria en 2018, y consiguiendo en 2020 la emisión de 566 millones de Documentos Tributarios Electrónicos (Gil, 2022).

Barreix y Zambrano (2018) han publicado experiencias de algunos países latinoamericanos que demuestran la eficacia de la factura electrónica frente a la factura tradicional en papel tales como: el desarrollo del *factoring*, en Chile; el seguimiento de las mercancías transportadas, en Brasil; y la identificación y análisis de productos nacionales en los encadenamientos productivos, en Ecuador

La facturación electrónica se ha convertido, sin duda alguna, en un elemento esencial en el proceso de transformación digital de los países de Latinoamérica por los beneficios que supone y por la efectividad comprobada.

Facturación Digital en Cuba

La pandemia de la Covid – 19, que azota al mundo desde el 2020, ha obligado a los gobiernos a adoptar medidas de aislamiento y limitación de la movilidad, hasta donde sea factible sin paralizar las actividades económicas. Derivado de ello el gobierno cubano orientó medidas para detener la transmisión de esta enfermedad, entre ellas la limitación del tránsito de personas y vehículos, fundamentalmente entre municipios y provincias. Este entorno dificultó la gestión para obtener la

firma del comprador en la factura emitida por el vendedor, lo que conllevó más días para emitir y contabilizar la factura y reconocerse el ingreso en la información financiera del vendedor. En este contexto la factura digital se inserta dentro del comercio electrónico que se impone inevitablemente como parte del proceso de informatización a escala mundial, en el cual Cuba se ha propuesto avanzar pese a las dificultades que afronta, con miras a ubicarse al nivel de los países que han desarrollado esta factura de nuevo tipo y sus consiguientes beneficios. Su empleo fue estatuido por el Ministerio de Finanzas y Precios (2021), (Martinez-Marsal & Guia-Alcolea, 2021).

La infraestructura tecnológica en el ámbito empresarial cubano tuvo un crecimiento positivo en los últimos años, siempre en los marcos de la soberanía tecnológica. Se ha logrado una accesibilidad a la red de redes considerable, aunque no ha sido así en el sector del cliente minorista, el cual aún no cuenta con una conectividad cómoda, privada y eficiente a internet, de ahí que los primeros pasos deben ir dirigidos a la modalidad de comercio electrónico mayorista.

Existe un número considerable de empresas en el país que no han percibido aún los beneficios de muchas de estas tecnologías y desean hacerlo. Conocen por experiencias ajenas las oportunidades que le brindan para mejorar su eficiencia y obtener sustanciosos beneficios económicos

Descripción de los procesos que serán automatizados

En la empresa Aicros, que se dedica a la venta de servicios de *software* para el Ministerio de la Construcción y trabajadores por cuenta propia, es muy importante el tratamiento y nivel de respuesta al cliente, y dentro de los servicios que esta entidad brinda, la contratación y gestión de las prefacturas es uno de los que no se puede dejar pasar por alto si se quiere llegar a tener una calidad óptima en el servicio y a su vez cumplir con lo legislado en nuestro país en este tema. Para ello es imprescindible en la actualidad contar con una herramienta informática que controle este proceso, ya que se ha visto afectado debido a la cantidad de información que es necesario manejar con poco personal encargado de estas tareas y sin un sistema o solución informática para esto.

El proceso comienza cuando el cliente desea adquirir un producto. Para realizar esta acción se dirige a la tienda online que posee la entidad, donde podrá obtener el producto, pero no podrá recibir la licencia correspondiente al software adquirido hasta que no haya realizado el pago correspondiente, para lo cual es necesario haber concertado con anterioridad un contrato con la entidad. Para poder concertar un contrato con la entidad el cliente debe:

- Dirigirse a la sede de la MYPIME y reunirse con el jurídico. Este le informara de los distintos tipos de contratos que se pueden tener.
- Una vez el cliente elige el tipo de contrato acorde a su necesidad el jurídico imprimirá una plantilla de contrato y procederá a explicar los datos que necesitan para llenar la plantilla, así como los documentos adjuntos que se le adicionaran.
- Se concertará una cita para la firma del contrato, siempre y cuando los datos y archivos suministrados por el cliente sean correctos, si no se aplazara la firma hasta que se obtengan.
- Una vez se haya firmado el contrato con la entidad, sus datos se guardan en el sistema ODOO (ODOO Community, 2022) y pueden ser encontrados por el especialista comercial. A partir de este momento se puede comenzar a realizar las compras, obteniendo la correspondiente factura.

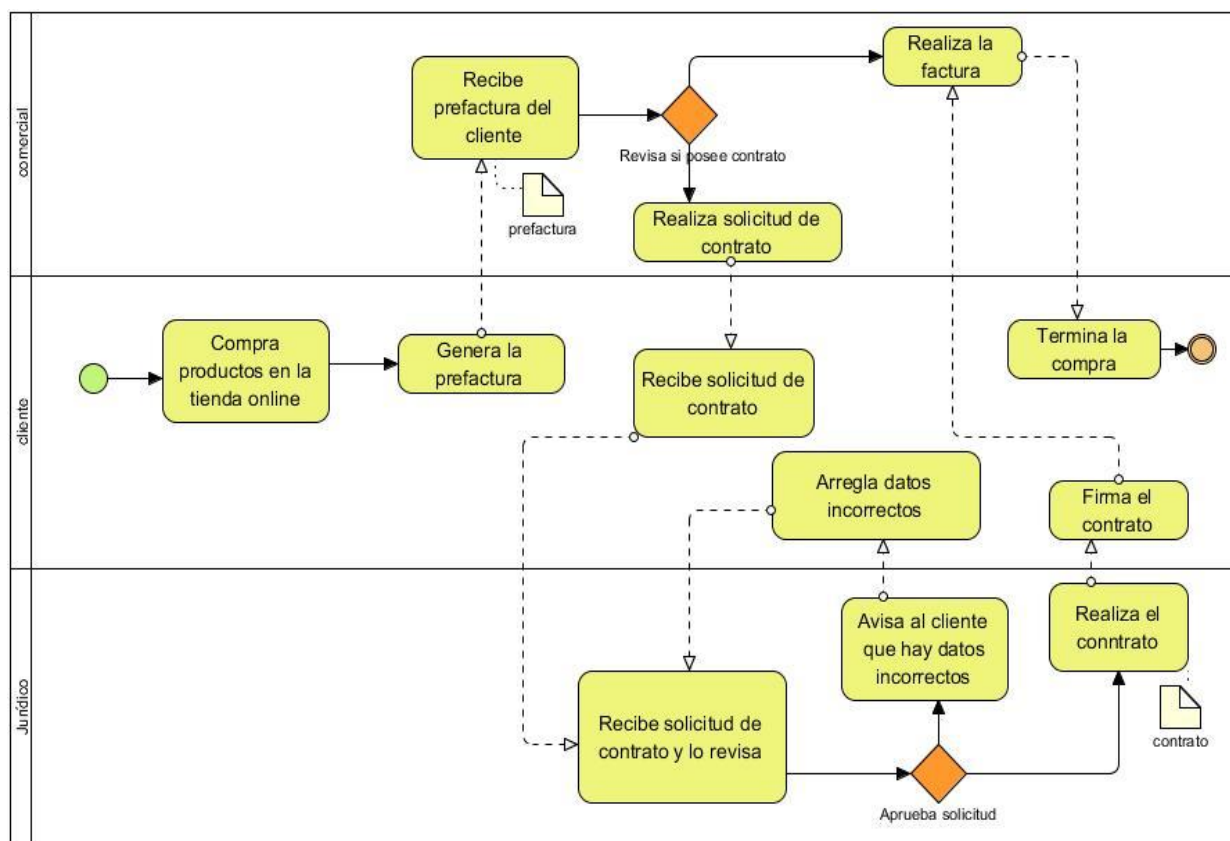


Figura 1.1. Flujo del proceso de comercialización. **Fuente** Aicros

.1.2 Análisis crítico de la ejecución de los procesos de contratación y prefacturación

Al realizarse la creación del contrato, de manera tradicional, se somete a los clientes a largas esperas para ser atendidos provocándoles incomodidades por tal motivo y por el detrimento de valiosa parte de su jornada laboral, se les dificulta la obtención de información inmediata y necesaria para la firma del contrato, los trabajadores que interactúan directamente con los

usuarios se encuentran saturados en esa labor conduciéndolos a ofrecer un servicio no óptimo y de cuestionable calidad.

Al realizarse la facturación en un sistema aparte, resulta difícil controlar la facturación de la tienda, y pueden aparecer errores al tener el comercial que estar trabajando simultáneamente con dos sistemas.

1.2. Análisis comparativo de propuestas existentes con la propuesta del trabajo.

Para el desarrollo de la investigación es necesario realizar el análisis de soluciones existentes que permitan establecer comparaciones con otros conocimientos paralelos, ofreciendo diferentes posibilidades de comprensión del tema a tratar. Para ello a continuación se realiza un análisis de los principales sistemas de gestión comercial existentes.

1.2.1 Sistema para la gestión de Oportunidades de Negocio para el Centro de Informática Industrial

El sistema es resultado del trabajo de diploma desarrollado por José Antonio López Martínez del Centro de Informática Industrial (CEDIN) en el año 2015, para ayudar a los especialistas del GIE de este centro a agilizar y controlar el proceso de gestión de oportunidades de negocio, como actividad inicial del proceso de gestión comercial. Las principales funcionalidades del sistema son (Martínez, 2015):

1. Autenticación de usuarios mediante el LDAP, insertar, modificar y eliminar usuarios y oportunidades.
2. Guardar la actualización de las oportunidades de acuerdo con los cambios que se vayan realizando.
3. Elaborar, modificar y eliminar ofertas comerciales.
4. Crear una base de cálculos asociada a una oportunidad de negocios.
5. Descargar los documentos asociados a una oportunidad, que se encuentren en el servidor en un archivo comprimido.
6. Informar vía correo electrónico, a los usuarios con roles jefe de centro y especialista comercial, cuando una oportunidad ha cambiado su estado.
7. Generar un reporte mensual con los datos de las oportunidades.

Se puede percibir que, aunque el sistema desarrollado tiene algunas de las funcionalidades necesarias para la propuesta de solución, omite algunas de las funcionalidades con amplias potencialidades para el proceso de negociación y contratación que se realiza en la empresa. No exporta los documentos a PDF. Además, no contribuye a mejorar la toma de decisiones de los asesores comerciales pues no genera reportes a partir de criterios de impacto para el proceso comercial en el centro.

1.2.2. Myros

Desarrollado por la empresa *Desoft*, *Myros* es un *software* que permite el control de la regulación de las relaciones contractuales que se establecen en la economía nacional entre los diferentes operadores que en ella intervienen. Entre sus principales funcionalidades se encuentran:

- Registrar contratos según su tipo: compra, venta, adquisición y prestación de servicios.
- Registrar mediante ficha de clientes las personas autorizadas a firmar contratos, facturas y otras operaciones.
- Emitir reportes del expediente de actividades contractuales de clientes y proveedores.
- Emitir reportes del control de la contratación según los días de vencimiento y vigencia de estos.
- Comunicar vía correo electrónico cuando el contrato está en término de expedición a la contraparte.

Este sistema aborda todo lo referente al proceso de contratación, pero no ofrece una solución a la necesidad de llevar un control de las facturas realizadas en la tienda, además al ser una aplicación aparte no resuelve el problema que surge al utilizar 2 aplicaciones simultáneamente.

1.2.3. Aportes de los sistemas homólogos estudiados.

Luego del estudio realizado se puede concluir que, a pesar de ser sistemas de gestión comercial, no describen un flujo completo del proceso llevado a cabo en el centro, por lo que no se ajustan en su totalidad a las necesidades reales de la entidad objeto de estudio. No obstante, su estudio ha tributado un conjunto de características relevantes para la realización de la propuesta de solución. De una manera u otra, han aportado una visión futura de lo que podría llegar a hacer el sistema que se propone, pero se hace evidente la necesidad de desarrollar un módulo a la tienda online existente que satisfaga todas las necesidades de la empresa.

1.3 Marco legal para la contratación en Cuba.

Existen reglas y principios en materia de contratos, que deben ser expresamente reconocidos de manera que pueda ser exigida su observancia desde el proceso mismo de concertación,

asegurando su mayor transparencia y contribuyendo a promover y asegurar las relaciones de cooperación entre las partes. Asimismo, se hace necesario dotar al sistema de contratación de la necesaria flexibilidad, de forma tal que alcance a comprender y tutelar, como plenamente eficaces, aquellas otras transacciones que tienen lugar en el ámbito de las relaciones interempresariales, sin que para ello se exija como presupuesto el otorgamiento de un contrato por escrito.

En Cuba la contratación económica se rige por el DECRETO-LEY No. 304 “DE LA CONTRATACIÓN ECONÓMICA” (Ministerio de Justicia, 2013).

En este se establece que durante la etapa de negociación las partes deben acreditar su personalidad y capacidad jurídicas y, cuando corresponda, la representación y también los documentos de creación o constitución de dichas partes y de la inscripción en el registro público correspondiente que, le otorga personalidad jurídica. El representante de cada parte en un contrato acredita su condición mostrando el documento justificativo de la representación.

Además, establece que el contrato puede ser verbal o escrito. Como regla debe ser escrito, bien sea manuscrito, en forma documental impresa o en soporte electrónico. Cuando las condiciones o características así lo aconsejen, las partes pueden concertar contratos verbales, observando en lo pertinente las reglas establecidas con carácter general para su concertación

1.4 Metodología y herramientas seleccionadas en la implementación de la solución.

La selección de las herramientas y tecnologías adecuadas para el desarrollo de la solución es un paso fundamental en el desarrollo de la investigación. A continuación, se realiza un estudio sobre las características principales asociadas a diferentes herramientas y tecnologías para seleccionar las más apropiadas para cumplir con el objetivo general del trabajo de manera que satisfaga al cliente.

1.4.1 Marcos de trabajo (*Frameworks*)

Sin lugar a duda, los marcos de desarrollo web se han convertido en uno de los principales componentes para el desarrollo de aplicaciones web en la actualidad. Los *frameworks* simplifican el desarrollo de una aplicación mediante la automatización de algunos patrones utilizados para resolver tareas comunes. Proporcionan estructura al código fuente y encapsulan operaciones complejas en instrucciones sencillas que obligan al desarrollador a crear un código más legible y fácil de mantener. Para la presente investigación se seleccionaron los frameworks Nest, js y Angular teniendo en cuenta su amplio uso en la línea de desarrollo web en la empresa Aicros.

1.4.1.1 *Angular*.

Angular es un *framework* de *JavaScript* de tipo *open source*, es decir de código abierto, desarrollado por *Google*. Se utiliza en la creación y programación de aplicaciones *web* de una sola página.

Cuando se creó en 2010 se llamaba *AngularJs*. Al poco tiempo este *framework* se hizo muy popular y comenzó a llamarse simplemente *Angular*. Al momento de presentarse *Angular2*, prácticamente la totalidad del código fue reescrito en *TypeScript*, haciendo énfasis en el desarrollo de dispositivos móviles. La meta era que los programadores pudieran utilizar un código más estructurado y reutilizable (Cristancho, 2022).

¿Por qué *Angular*?

1. *Angular* cuenta con una gran serie de módulos habituales para el desarrollo de proyectos *web*, por lo que no es necesario empezar desde cero. Entre sus muchas otras ventajas se encuentra (¿Qué es *Angular*? Características y ventajas, 2022):
2. Alta calidad de la aplicación: Si bien este es un lenguaje difícil de aprender para desarrolladores *web* principiantes, a la vez representa una gran ventaja, porque el éxito del producto está prácticamente asegurado. Se puede crear cualquier cosa con las funciones integradas del marco de trabajo.
3. Proceso de desarrollo *web* más rápido: Permite crear aplicaciones de forma más rápida y eficiente, pues goza de ventajas técnicas como pueden ser:
 - Documentación detallada: La documentación está cuidadosamente escrita y dotada de una gran variedad de ejemplos de código para tener mayor claridad en el proceso y permitir que los desarrolladores encuentren soluciones rápidas a cualquier problema conforme crean una aplicación.
 - Interfaz de línea de comando en *Angular*: La *CLI* (por sus siglas en inglés) facilita el trabajo de los desarrolladores al ofrecer un conjunto de herramientas de codificación. Además, esta línea de comandos puede ampliarse con bibliotecas de terceros para resolver problemas de software inusuales o muy complicados.
 - Soporte de *Google*: *Angular* es mantenido por *Google*, por lo que recibe soporte en tiempo y forma a cualquier duda o problema que se pueda presentar.
4. Aplicaciones *web* más ligeras: Versiones anteriores de *Angular* presentaban problemas en el tamaño de las aplicaciones, lo que no permitía una carga rápida. Actualmente se han hecho mejoras en los módulos de carga diferida y con el redendizador de *Ivy*, que se permite hacer paquetes más pequeños para acelerar la aplicación.
5. Código legible y comprobable: Podemos resaltar que *Angular* es un marco de trabajo consistente gracias a sus elementos estructurales como módulos, componentes, directivas,

pipas y servicios; así como la posibilidad de escribir aplicaciones con la arquitectura tradicional MVVM (Modelo-Vista-Vista-Modelo) y MVC (Modelo-Vista-Controlador). Ambas sirven para el mejoramiento de reutilización de código.

1.4.1.2 NestJS.

NestJS es un *framework* para desarrollo de aplicaciones del lado del servidor escalables y mantenibles, sobre la plataforma *NodeJS*. Este *framework* permite desarrollar aplicaciones *Backend* usando la plataforma *NodeJS*, es decir, mediante el lenguaje *JavaScript* para el servidor. Además *NestJS* permite usar también *TypeScript*, para mejorar todavía más las prestaciones para el desarrollador (desarrolloweb, 2021).

¿Por qué NestJS?

Node es una plataforma de ejecución de *JavaScript* bastante minimalista. Básicamente se trata de un intérprete de *JavaScript* que permite ejecutar programas de consola, con propósito general. Algunos *frameworks* de *NodeJS*, como *Express*, permiten realizar aplicaciones del lado del servidor, capaces de responder a solicitudes *HTTP*. *Express* se usa mucho para construir servicios web, como es el caso de *APIs REST*.

Express y otros *frameworks NodeJS* están muy bien, lo que ocurre es que dejan tradicionalmente muy suelto al desarrollador en lo que respecta a la arquitectura. Por ello el desarrollador acaba necesitando instalar diversas librerías, creando una estructura de carpetas que le sirva para sus necesidades, configurar las diversas herramientas, etc. *NestJS* a diferencia de otros *frameworks* en *Node* tiene un foco en la arquitectura. Es decir, entrega ya un proyecto de base y unas herramientas configuradas que nos permiten evitar mucho del trabajo inicial de una aplicación en *NodeJS*. Además, se trata de una arquitectura opinada, que nos garantiza buenas prácticas, mayor homogeneidad a los proyectos de las empresas y unas guías claras para los equipos de desarrollo. Otro detalle que considerar sobre la arquitectura propuesta por *Nest* es su fuerte inspiración en Angular, por lo que resultará familiar para una buena comunidad de desarrolladores que ya conocen este *framework* (Nestjs, 2022).

¿Qué ofrece NestJS?

Como hemos dicho, *Nest* nos entrega todo el marco sobre el que vamos a desarrollar una aplicación. Gracias al *framework* tienes ya configurada la compilación de *TypeScript* y una serie de bibliotecas básicas. Además, hace un uso intensivo de Programación Orientada a Objetos, lo que nos permite realizar un buen diseño de las aplicaciones.

Por debajo *NestJS* sigue trabajando con herramientas sólidas y conocidas del ecosistema *NodeJS*, como *Express* (y opcionalmente *Fastify*). Además, permite trabajar con cualquier librería

disponible en *NodeJS* que un proyecto necesite. Otra de las ventajas de *NestJS* es que tiene su propio *CLI*, que nos permite crear cómodamente nuevos proyectos y realizar *scaffolding* (técnica de generación de proyecto) de código por medio de comandos de consola, ejecutar cómodamente las aplicaciones y más.

En resumen, *NestJS* tiene como objetivo liberar al desarrollador de tareas comunes y repetitivas, ofrecer una arquitectura sólida para aplicaciones mantenibles y permitir preocuparse más del desarrollo y menos del *setup* de los proyectos (Álvarez M., 2020).

1.4.2 Lenguajes soportados por el *framework*

1.4.2.1 HTML5

Es un lenguaje de marcación que sirve para definir el contenido de las páginas *web*. Se compone en base a etiquetas, también llamadas marcas o tags, con las cuales se consigue expresar las partes de un documento, cabecera, cuerpo, encabezado, párrafos etc. En definitiva, el contenido de una página *web*.

Es el primer lenguaje que debe aprender cualquier persona en construir un sitio *web*. A partir de HTML podemos pasar a muchos otros lenguajes interesantes que sirven para hacer cosas diversas y más avanzadas (Desarrollo web, 2021).

Entre las principales ventajas que aporta el uso de este lenguaje de programación se encuentran (Sharp, 2011):

- Permite describir hipertexto.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web.
- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.

1.4.2.2 CSS3

Es el lenguaje de estilos utilizados para describir la presentación de documentos *HTML*. Describe como debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios.

CSS es uno de los lenguajes bases de la *Open Web* y posee una especificación estandarizada por parte del W3C. Es utilizado para diseñar y dar estilo a las páginas *web*, por ejemplo, alterando

la fuente, el color, tamaño y espacio del contenido, dividirlo en múltiples columnas o agregarle animaciones y otras características decorativas (Desarrollo web).

1.4.2.3 TypeScript

Es un lenguaje de programación fuertemente tipado que se basa en *JavaScript*, lo que le brinda mejores herramientas a cualquier escala. Es de código abierto desarrollado y mantenido por *Microsoft*. Extiende la sintaxis de *JavaScript*, por tanto, cualquier código *JavaScript* existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de *TypeScript* se traduce a código *JavaScript* original (Microsoft, 2022).

1.4.3 Entorno de desarrollo integrado (IDE):

Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con *Git*, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación.

Para tener una idea de la popularidad de *Visual Studio Code* y la aceptación que ha tenido en el mundo de desarrollo, según una encuesta realizada por *Stack Overflow* a más de 80,000 desarrolladores en mayo del 2021, *Visual Studio Code* es el entorno de desarrollo más usado y con mucha diferencia, un 71.06% (Flores, 2022).

La gran customización y la amplia gama de funcionalidades que se le pueden añadir, permiten al desarrollador crear un entorno de trabajo agradable y útil, mejorando la velocidad de desarrollo y la comodidad del programador.

1.4.4 Visual Paradigm

El **Visual Paradigm** es una *suite* completa de herramientas CASE que utiliza el Lenguaje Unificado de Modelado (*UML*, por sus siglas en inglés), permitiendo el soporte del ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Visual Paradigm 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. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: *Enterprise*, *Professional*, *Community*, *Standard*, *Modeler* y *Personal*. Existe una alternativa libre y gratuita de este *software*, la versión *Visual Paradigm UML 6.4 Community*

Edition (Community Edition, ya que existe la Enterprise, Professional, etc.). Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos.

Se caracteriza por. (Visual Paradigm Internacional, n.d.):

1. Disponibilidad en múltiples plataformas (*Windows, Linux*).
2. Diseño centrado en casos de uso y enfocado al negocio que generan un *software* de mayor calidad.
3. Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
4. Capacidades de ingeniería directa e inversa.
5. Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
6. Disponibilidad de múltiples versiones, con diferentes especificaciones.
7. Licencia: gratuita y comercial.
8. Soporta aplicaciones *Web*.
9. Las imágenes y reportes generados no son de muy buena calidad.
10. Varios idiomas.
11. Generación de código para Java y exportación como HTML.
12. Fácil de instalar y actualizar.
13. Compatibilidad entre ediciones.
14. Soporte de UML versión 2.1.
15. Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
16. Modelado colaborativo con CVS y Subversión (control de versiones).
17. Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI.
18. Ingeniería de ida y vuelta.
19. Ingeniería inversa - Código a modelo, código a diagrama.
20. Ingeniería inversa *Java, C++, Esquemas XML, XML, NET exe/dll, CORBA IDL*.
21. Generación de código - Modelo a código, diagrama a código.

22. Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
23. Generación de código y despliegue de EJB - Generación de *beans* para el desarrollo y despliegue de aplicaciones.
24. Diagramas de flujo de datos.
25. Soporte ORM - Generación de objetos Java desde bases de datos.
26. Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
27. Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
28. Generador de informes.
29. Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
30. Importación y exportación de ficheros XML.
31. Integración con *Visio* - Dibujo de diagramas UML con plantillas (*stencils*) de *Microsoft Visio*.
32. Editor de figuras.

1.4.5 Gestores de Bases de datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipularlos, garantizando su seguridad e integridad (Álvarez S., 2017).

Se basa en *Electron*, un *framework* que se utiliza para implementar *Chorium* y *NodeJS* como aplicaciones para escritorio, que se ejecuta en el motor de diseño *Blink*. Aunque utiliza el *framework Electron*, el *software* no usa *Atom* y en su lugar emplea el mismo componente editor (Mónaco) utilizado en *Visual Studio Team Services* (anteriormente llamado *Visual Studio Online*). (Microsoft, 2022).

1.4.5.1 *Postgres SQL*.

Postgres SQL es un sistema de gestión de bases de datos relacionales de objetos (ORDBMS) basado en *postgres*, versión 4.21 (Grupo de desarrollo global de PostgreSQL, 2022),

desarrollado en el departamento de ciencias de la computación de la Universidad de California en *Berkeley*. Fue pionero en muchos conceptos que solo estuvieron disponibles en algunos sistemas de bases de datos comerciales mucho más tarde.

Postgres SQL es un descendiente de este código abierto original de *Berkeley*. Soporta gran parte del estándar *SQL* y ofrece muchas características modernas:

- Consultas complejas.
- Llaves extranjeras.
- *Triggers*.
- *Views*.
- *Transaccional integrity*.
- Control de concurrencias multiversión.

Además, el usuario puede ampliar *PostgreSQL* de muchas maneras:

- Agregando nuevos tipos de datos.
- Funciones.
- Operadores.
- Funciones agregadas.
- Métodos de índice (The PostgreSQL Global Development Group, 2010).

1.4.3 Servidor Web: *Node*

El servidor seleccionado para la aplicación es *Node*, que tiene como características más sobresalientes:

Multiplataforma: Se puede ejecutar tanto en sistemas *Windows*, *MacOS*, *Linux*, independientemente de la arquitectura que use la máquina.

Asíncrono y controlado por eventos: Un servidor basado en *Node* nunca espera que una API devuelva datos. El servidor pasa a la siguiente API después de llamarlo y un mecanismo de notificación de eventos de *Node* ayuda al servidor ayuda a obtener una respuesta de la llamada API anterior.

Rapidez: Al estar construido en el motor de *JavaScript V8* de *Google Chrome*, la biblioteca de *Node.js* es muy rápida en la ejecución de código.

Sin almacenamiento en *buffer*: Las aplicaciones de *Node* nunca almacenan en *buffer* ningún dato, estas aplicaciones simplemente generan los datos en fragmentos.

Procesos de un solo hilo, pero altamente escalable: *Node.js* utiliza un modelo de un solo hilo con bucle de eventos. El mecanismo de eventos ayuda al servidor a responder sin bloqueos y hace que el servidor sea altamente escalable en comparación con los servidores tradicionales que crean hilos limitados para manejar las solicitudes. *Node.js* utiliza un solo programa de subprocesos y el mismo programa puede proporcionar un servicio a un número mucho mayor de solicitudes que los servidores tradicionales como el servidor Apache.

Licencia Basada en código abierto: *Node.js* se distribuye gracias a una licencia del MIT (NTT DATA, 2020).

1.4.4 Sistema de control de versiones *Git*

Git es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes. Con rapidez y eficiencia.

Es fácil de aprender y ocupa poco espacio con un rendimiento ultrarrápido. Supera las herramientas de SCM como *Subversion*, *CVS*, *Perforce* y *ClearCase* con características como sucursales locales económicas, áreas de preparación convenientes y múltiples flujos de trabajo.

La principal característica de *Git* que la diferencia de casi todos los demás SCM es su modelo de ramificación. *Git* lo permite y lo alienta a través de varias sucursales locales que pueden ser completamente independiente entre sí. La creación, fusión y eliminación de esas líneas de desarrollo lleva unos segundos.

Eso significa que se pueden hacer cosas como:

- Cambio de contexto sin fricción.
- Líneas de código basada en roles.
- Flujo de trabajo basado en características.
- Experimentación desechable.

Con *Git* casi todas las operaciones se realizan localmente, lo que le brinda una gran ventaja de velocidad en los sistemas centralizados que constantemente tienen que comunicarse con un servidor en alguna parte. Fue creado para funcionar en el *kernel* de *Linux*, lo que significa que ha tenido que manejar de manera efectiva grandes repositorios desde el primer día. La velocidad y el rendimiento han sido un objetivo de diseño principal de *Git* desde el principio.

Git se publica bajo la licencia publica general GNU versión 2.0, que es una licencia de código abierto. El proyecto *Git* eligió utilizar GPLv2 para garantizar su libertad para compartir y cambiar el *software* libre, para asegurarse de que el *software* sea gratuito para todos los usuarios (Git, 2022).

1.4.6 Metodología de desarrollo

El éxito de un proyecto está directamente relacionado con la metodología que se utiliza durante su desarrollo. Las metodologías de desarrollo de software son un conjunto de procedimientos y técnicas para la implementación de aplicaciones que tienen como línea fundamental definir ordenadamente un conjunto de tareas a realizar. Existen metodologías de desarrollo ágiles y prescriptivas (o pesadas) como también se conocen.

1.4.6.1 Programación Extrema (XP)

Dentro de las metodologías ágiles más utilizadas se encuentra XP. Esta metodología se centra en potenciar las relaciones interpersonales haciendo de estas el punto clave para el éxito del proyecto. Por otra parte, promueve el trabajo en equipo propiciando un clima excelente en el ámbito laboral y permitiendo un ambiente de constante aprendizaje para los desarrolladores. Continuamente se encuentra retroalimentándose de la relación cliente y el equipo de desarrollo, de las relaciones internas entre cada uno de los individuos y de la sencillez de las soluciones que se vayan implementando (Joskowicz., 2008).

Brinda además seguridad para la gestión de cambios, ya que está orientado fundamentalmente para aquellos proyectos de requisitos imprecisos con altas probabilidades de cambio y donde el riesgo técnico sea elevado. Se recomienda también para proyectos donde el cliente forme parte del equipo de desarrollo, o al menos esté lo suficiente como para aclarar los cambios y dudas en el negocio (Joskowicz., 2008).

La programación extrema engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas. La Figura 1.2 muestra cada una de las actividades de XP, y resalta las tareas claves de cada una (Pressman, 2010):

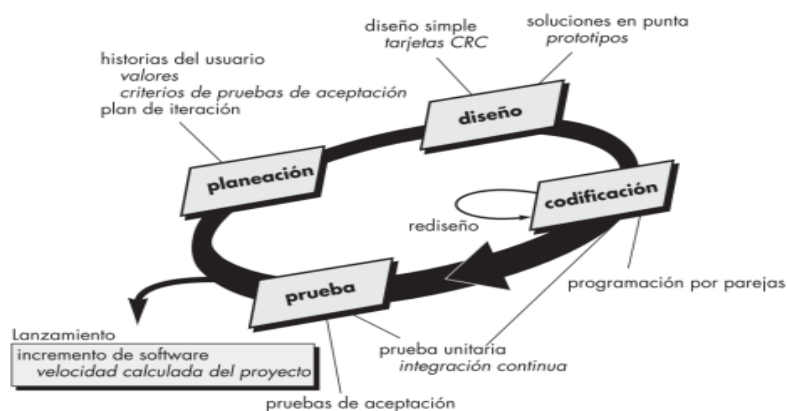


Figura 1.2 El proceso de la programación extrema.

Fuente: Pressman (2010)

Planeación: Esta actividad comienza escuchando a los clientes, para entender el contexto del negocio y definir las características principales y funcionalidad que se requiere, estas características se transforman en requerimientos del negocio que se especifican mediante Historias de Usuario; las cuales recogen la interacción hablada entre desarrolladores y usuarios. Una vez hechas las Historias de Usuario, el equipo de desarrollo las divide en tareas, estima el esfuerzo, recursos requeridos para su implementación, se genera el plan de entregas, las iteraciones, la rotación de parejas y las reuniones diarias.

Diseño: Es la etapa en donde son evaluadas las historias de usuario por el equipo del proyecto para dividir las en tareas, cada tarea representa una característica distinta del sistema y se puede diseñar una prueba de unidad que verifique cada tarea, estas tareas se representan por medio de las tarjetas CRC (Clase-Responsabilidad-Colaborador). Las tarjetas CRC identifican y organizan las clases bajo el paradigma orientado a objetos (lo que incluye asignación de responsabilidades), cada tarjeta contiene el nombre de la clase (que representa una o más historias de usuario), una descripción de las responsabilidades o métodos asociados con la clase, así como la lista de las clases con que se relaciona o que colaboran con ella. Las tarjetas CRC son el único trabajo de diseño que se genera como parte del proceso de XP.

Codificación: Se lleva a cabo la programación en pareja, la unidad de pruebas y la integración del código. Durante esta etapa se espera la disponibilidad del cliente para que éste pueda resolver cualquier duda que se presente durante una jornada de trabajo.

Prueba: Cada tarea que se identificó con las historias de usuario, representa una característica distinta del sistema y se realiza una prueba de unidad por cada una de ellas, existen pruebas unitarias las cuales son diseñadas para probar cada uno de los métodos y clases, dichas pruebas son realizadas por los programadores.

1.5 Tendencias tecnológicas a considerar

Arquitectura N capas

La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Un buen ejemplo de este método de programación sería el modelo de interconexión de sistemas abiertos. Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo,

cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles. En el diseño de sistemas informáticos actual se suelen usar las arquitecturas multinivel o Programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). El diseño más utilizado actualmente es el diseño en tres niveles (o en tres capas).

Capas y niveles:

1. Capa de presentación: es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.

2. Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

3. Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un único computador, si bien lo más usual es que haya una multitud de computadoras en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo computador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más computadoras. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varias computadoras las cuales recibirán las peticiones del computador en que resida la capa de negocio. Si, por el contrario, fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en uno o más computadores que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de computadores sobre los cuales corre la capa de negocio, y otra serie de computadores sobre los cuales corre la base de datos.

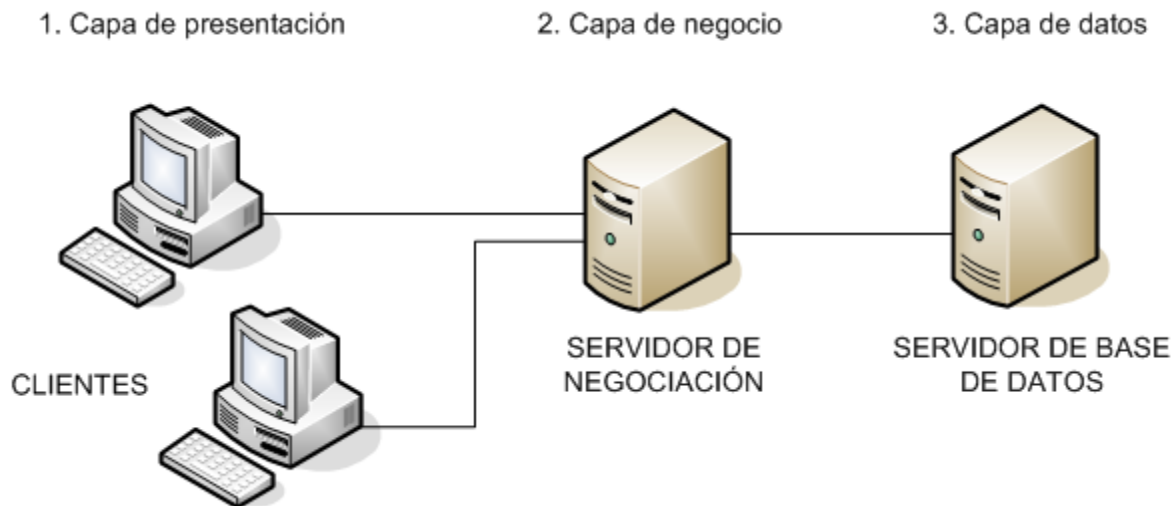


Figura # 1.3. Capas y Niveles.

Fuente: (Sánchez, 2017)

Arquitectura de 2 niveles

La arquitectura en 2 niveles se utiliza para describir los sistemas cliente/servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud, con sus propios recursos. Esto significa que el servidor no requiere otra aplicación para proporcionar parte del servicio.

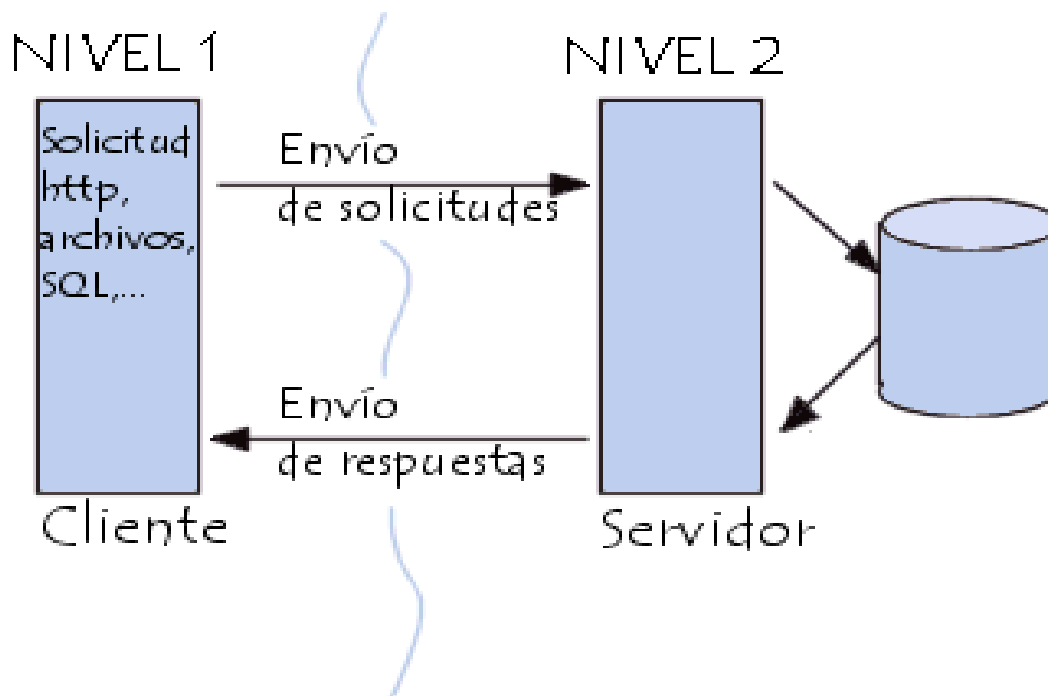


Figura # 1.4. Arquitectura en 2 niveles.

Fuente: (Sánchez, 2017)

Arquitectura en 3 niveles

En la arquitectura en 3 niveles, existe un nivel intermediario. Esto significa que la arquitectura generalmente está compartida por: Un cliente, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador *Web*) para la presentación. El servidor de aplicaciones (también denominado *software* intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo. El servidor de datos, que proporciona al servidor de aplicaciones los datos que requiere. Comparación entre ambos tipos de arquitecturas La arquitectura en 2 niveles es, por lo tanto, una arquitectura cliente/servidor en la que el servidor es polivalente, es decir, puede responder directamente a todas las solicitudes de recursos del cliente. Sin embargo, en la arquitectura en 3 niveles, las aplicaciones al nivel del servidor son descentralizadas de uno a otro, es decir, cada servidor se especializa en una determinada tarea, (por ejemplo: servidor web/servidor de bases de datos). La arquitectura en 3 niveles permite: Un mayor grado de flexibilidad, mayor seguridad, ya que la seguridad se puede definir independientemente para cada servicio y en cada nivel mejor rendimiento, ya que las tareas se comparten entre servidores. (Sánchez, 2017).

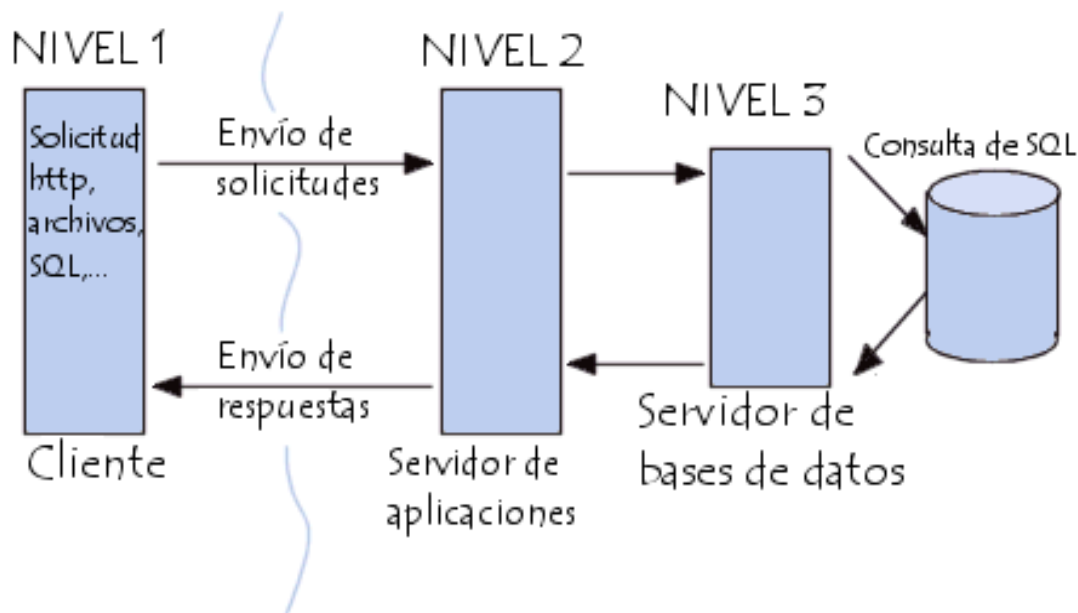


Figura # 1.5. Arquitectura en 3 niveles.

Fuente: (Sánchez, 2017).

Patrón de diseño Modelo Vista Controlador

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, entre otras). El modelo se encarga de la abstracción de la lógica

relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación (Guerra Velázquez, (2013)):

- ✓ El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- ✓ La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- ✓ El controlador se encarga de procesar las interacciones del usuario, manejo de las peticiones, de la seguridad, cargar la configuración de la aplicación y realiza los cambios apropiados en el modelo o en la vista.

Este patrón de diseño al separar los distintos componentes de la aplicación permite un más fácil mantenimiento y una mejor escalabilidad de la aplicación, aspectos que se quieren lograr en el presente proyecto.

Conclusiones.

En este Capítulo se han sentado las bases para la elaboración de la propuesta de investigación, el desarrollo del trabajo y la presentación de los resultados. Con la elaboración del mismo se ha demostrado que la empresa Aicros realmente necesita un sistema automatizado que controle el proceso de contratación y gestión de las prefecturas que le brinda a sus clientes, se evidencia además que un módulo agregado a su tienda *online* es la solución más adecuada para desarrollar un sistema que gane en eficiencia. Los lenguajes de programación más adecuados para la realización de la aplicación por las ventajas antes mencionadas son *Angular* y *NestJs*, con *Postgres* como gestor de base de datos y *XP* como metodología de desarrollo.

Capítulo 2 Descripción de la solución propuesta.

Introducción.

Para el correcto desarrollo de un *software* es necesario un buen diseño previo y tener claridad sobre lo que se quiere hacer, por eso una vez definido a lo que se pretende llegar se hace necesario la modelación del sistema a través de las Historias de Usuarios (HU) que acumulan la necesidad existente definida por el cliente, es llevado a cabo el análisis de los requerimientos. Se aplica la Metodología XP, *Extreme Programming* (Programación Extrema), metodología ágil de desarrollo, con el objetivo de garantizar el diseño de un programa lo más ajustado posible y se logra como ventaja la incorporación del cliente como un miembro del equipo de desarrollo.

También se realiza el estudio de factibilidad correspondiente a la implementación del sistema basado en el modelo Puntos de Función, mostrándose la planificación del sistema con sus cálculos correspondientes y los beneficios tangibles e intangibles.

2.1 Descripción de la solución

Se propone desarrollar una aplicación *web* que permita informatizar el proceso de comercialización de la Empresa de Informática y Automatización para la Construcción, AICROS, logrando una mayor eficiencia de este proceso de ventas y permita un mayor control sobre la facturación de sus productos y servicios, proporcionando a sus directivos más información para la toma de decisiones efectiva.

Los usuarios que interactúan con la aplicación tendrán los siguientes roles:

- Administrador del sistema: es el usuario que tiene la responsabilidad de usuarios internos, asignarles permisos, configurar y auditar la aplicación.
- Cliente o Representante: es el usuario registrado que interactúa mayormente con el sistema, crea contratos y lista los contratos que estén a su nombre.
- Jurídico: es el usuario que puede crear contratos a cualquier cliente, es el encargado de asignar un número serial al contrato y validarlo una vez se haya firmado, listar todos los contratos y auditarlos.
- Comercial: es el usuario que puede listar las prefacturas de compra de los clientes y después de que se efectúa el pago pasarla a estado facturado y auditarlos

Para listar los contratos y prefacturas, se hace uso de una plantilla que los muestra con cierto orden y clasificación, y ofrece facilidades de búsqueda a partir de la aplicación de filtros.

2.2 Etapa de planificación

La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando “Historias de usuarios”, las que sustituyen a los tradicionales “casos de uso”. Una vez obtenidas las “Historias de Usuarios”, los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Si alguna de ellas tiene “riesgos” que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba, para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un plan o cronograma de entregas en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, en dónde en cada una de ellas se desarrolla, prueba e instala unas pocas “historias de usuarios”

Figura 2.1 Fases de la metodología XP.



Figura 2.1 Fases de la metodología XP **Fuente:** Kent & Andres (2004).

2.2.1 Equipo de trabajo y roles

La metodología XP define roles de trabajo asociando a cada uno con diversas actividades. A continuación, se definen los roles, quedando designado el programador que sería el encargado de producir el código del sistema, el jefe de proyecto y el cliente que no es más que el que escribe las historias de usuario, les asigna la prioridad y diseña las pruebas funcionales para validar su implementación. A continuación, se muestra la asignación de estos roles a las personas responsables.

Tabla 2.1 Equipo de trabajo y roles.

Miembro	Roles
Mayli Estopiñán Lantigua	Jefe de proyecto, Tester
Roxana Naranjo Benítez	Programador
Avelino Martínez	Cliente

Fuente: Elaboración Propia

2.2.2 Historias de Usuario Iniciales

Las “Historias de usuarios” sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios. Son utilizadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si la aplicación cumple con lo que especifica la historia de usuario (Laínez Fuentes, 2017).

Escalas equivalentes a la prioridad en el negocio:

- Alta: Asignada a las Historias de Usuario que corresponden a funcionalidades esenciales en el desarrollo del proyecto, a las que el cliente define como primordiales.
- Media: Dada a las Historias de Usuario que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación directa sobre el proyecto que se esté desarrollando.

- Baja: Se le otorga a las Historias de Usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el proyecto en desarrollo.

Escala Nominal de Riesgo en Desarrollo:

- Alta: Cuando para la implementación de la Historia de Usuario se considera la posible existencia de errores que lleven a inoperatividad del código.
- Media: Cuando pueden aparecer errores en la implementación de la Historia de Usuario que puedan retrasar la entrega de la versión.
- Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

A continuación, se colocan las Historias de Usuario.

Tabla 2.2 Planificación de las Historias de Usuario.

No	Nombre	Prioridad	Riesgo	Esfuerzo	Iteración	Entrega
1	Diseño y creación de la base de datos	Media	Alto	1	1	1
2	Diseño de la interfaz de usuario	Media	Medio	1		
3	Listar contratos por el cliente	Alta	Medio	1		
4	Gestionar contratos por el cliente	Alta	Alto	2	2	2
5	Listar contratos, jurídico	Alta	Alta	3		
6	Gestionar contratos, jurídico	Alta	Alta	2		
7	Generar contrato	Alta	Medio	3	3	3
8	Listar prefacturas	Media	Alta	2		
9	Validar prefacturas	Alta	Alta	2		
14	Generar reportes	Media	Medio	1		

Fuente: Elaboración Propia.

Una vez detectadas las historias de usuario es importante elaborar el plan de entrega y de iteraciones.

2.2.3 Plan de Iteraciones

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una

reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación.

El proyecto fue dividido en 3 iteraciones, por lo que se obtuvo un total de tres entregas para las cuales se desarrollaron partes de la aplicación completamente funcionales. Para la determinación de cada una de las iteraciones se tuvo en cuenta la opinión del cliente a través de las entrevistas que se le realizaron continuamente antes de comenzar a desarrollar cada iteración donde se tomaron todos los acuerdos necesarios. Una vez concluida la iteración que va a estar conformada por un conjunto de historias de usuarios, teniendo en cuenta los requisitos exigidos por el cliente, la aplicación poseerá mayor número de funcionalidades.

A continuación, se muestran las tres iteraciones y las semanas en las que se desarrollan.

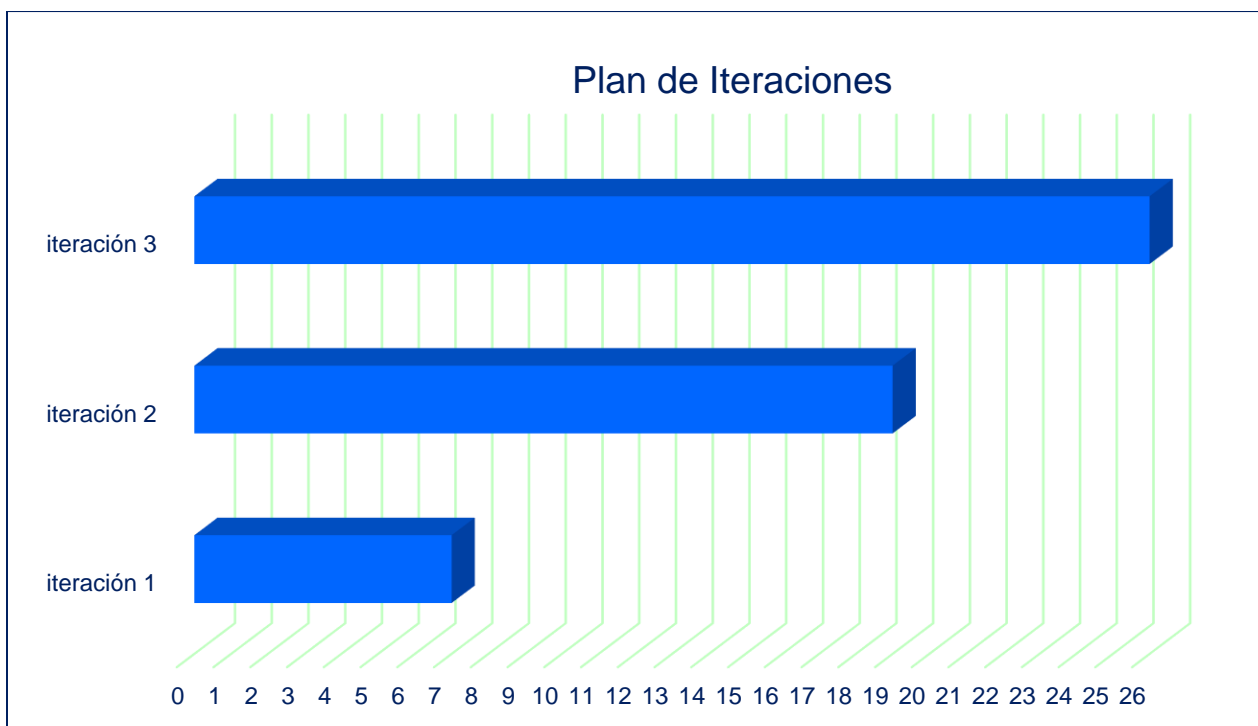


Figura 2.2 Plan de Iteraciones. **Fuente:** Elaboración Propia.

2.3 Planificación del proyecto.

Puntos de función: Es una métrica que permite traducir en un número el tamaño de la funcionalidad que brinda un producto de software desde el punto de vista del usuario, a través de una suma ponderada de las características del producto.

2.3.1 Componentes.

EI: Proceso en el que se introducen datos y que suponen la actualización de cualquier archivo interno

EO: Procesos en los que se envían datos al exterior de la aplicación.

EQ: Procesos consistentes en la combinación de una entrada y una salida en el que la entrada no produce ningún cambio en ningún archivo y la salida no contiene información derivada.

ILF: Grupo de datos relacionados entre sí internos al sistema.

EIF: Grupos de datos que se mantienen externamente.

Tabla 2.3. Componentes

Componente	Baja	Media	Alta
EI	2	2	4
EO	3	2	1
EQ	2	1	1
ILF	6	3	12
EIF	0	0	0

Fuente: Elaboración Propia

2.3.2 Puntos de función sin ajustar:

Los puntos de función sin ajustar (PFSA) se calcula como la suma de los productos de cada componente por su peso determinado en la tabla correspondiente.

PFTe: Total de puntos de función para las entradas del sistema.

PFTo: Total de puntos de función para las salidas del sistema.

PFTq: Total de puntos de función para las consultas del sistema.

PFTif: Total de puntos de función para los archivos internos del sistema.

PFTef: Total de puntos de función para los archivos externos del sistema.

Tabla 2.4. Puntos de función sin ajustar

Componente	Bajo	Medio	Alto	Total
EI	$2*3=6$	$2*4=8$	$4*6=24$	PFTe=38
EO	$3*4=12$	$2*5=10$	$1*7=7$	PFTo=29
EQ	$2*3=6$	$1*4=4$	$1*6=6$	PFTq=16
ILF	$6*7=42$	$3*10=30$	$12*15=180$	PFTif=252
EIF	$EFb*5=0$	$EFm*7=0$	$EFa*10=0$	PFTef=0
PFSA				335

Fuente: Elaboración Propia

2.3.3 Puntos de función ajustados

Tabla 2.5. Puntos de fusión ajustados

Nº de Factor	Nº de Factor	Valor 0..5
1	Comunicación de Datos	3
2	Proceso Distribuido	3
3	Objetivos de Rendimiento	3
4	Configuración de Explotación Compartida	1
5	Tasa de transacciones	2
6	Entrada de Datos en Línea	4
7	Eficiencia con el Usuario Final	3
8	Actualizaciones en Línea	1
9	Lógica de Proceso Interno Compleja	2
10	Reusabilidad del Código	3
11	Conversión e Instalación contempladas	2
12	Facilidad de Operación	4
13	Instalaciones Múltiples	1
14	Facilidad de Cambios	4
	Ajuste de Complejidad Técnica (ACT)	36

Fuente: Elaboración Propia

$$PFA=335*[0.65*[0.01*47]]=78.39$$

$$\text{Líneas de código} = 78.39 * 54 = 4233$$

$$\text{Esfuerzo horas/persona} = 78.39 / [1/8] \text{ horas/persona} = 627.12 \text{ horas/persona}$$

Duración del proyecto

Horas por persona: 627.12

Cantidad de personas trabajando en el proyecto: 1

Horas productivas: 20 días laborales y 5 horas productivas estimadas de las 8 de una jornada laboral diaria (100 horas productivas al mes)

$$\text{Duración del proyecto en horas} = 627.12 \text{ horas} / 1 \text{ persona} = 627.12$$

Duración del proyecto en meses= $627.12 / 100\text{horas/mes}$ = 6 meses y medio

Cálculo del presupuesto del proyecto

Sueldo del participante= 6895 CUP por mes

Otros costos necesarios= 2000 CUP

Costo total del proyecto= $6895 * 6.5 \text{ meses} + 5000 = \underline{46817.5}$

2.3.4 Análisis de Costos y Beneficios:

Este *software* traerá consigo un conjunto de beneficios tangibles e intangibles para los usuarios finales y desarrolladores.

Dentro de los **beneficios intangibles** que se esperan alcanzar se encuentran los asociados a la calidad del manejo de los datos relacionados con Contratación en el departamento Legal de la PyME tomada como punto de referencia, así como el ahorro de tiempo que implica el acceso a la información recogida en el sistema. Estos beneficios están asociados fundamentalmente a la existencia de un sistema que permita un fácil acceso y control estricto de la información relacionada con los equipos y partes en garantía.

Los **beneficios tangibles** se esperan a mediano plazo con la posibilidad de tomar decisiones en el control del proceso de contratación y la gestión de las prefacturas a partir de datos concretos, que pudiese estar dado, por ejemplo, en la cantidad de contratos realizados en una determinada cantidad de tiempo, las compras realizadas en el último año y los beneficios alcanzados en la entidad.

2.4 Plan de Entregas

El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por el desarrollador. Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario.

Tabla 2.6 Plan de entrega.

Entrega	Descripción
1	Esta entrega comprende:
	- Diseño y creación de la base de datos.
	- Diseño de la interfaz de usuario.
	- Listar contratos por el cliente
	- Gestionar contratos cliente.
2	Esta entrega comprende:
	- Listar contratos por el jurídico.
	- Gestionar contratos por el jurídico.
	- Generar contrato.
3	Esta entrega comprende:
	- Listar prefacturas.
	- Validar prefacturas.
	- Generar reportes.

Fuente: Elaboración Propia



Figura 2.3 Plan de entregas.

Fuente: Elaboración Propia.

2.5 Etapa de diseño

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible que a la larga costará menos tiempo y esfuerzo desarrollar. Si surgen problemas potenciales durante el diseño, XP sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema (Pressman, 2010).

2.5.1 Prototipo de la interfaz de usuario

Se muestran ilustraciones de la interfaz de usuario en la figura 2.4 a la figura 2.7

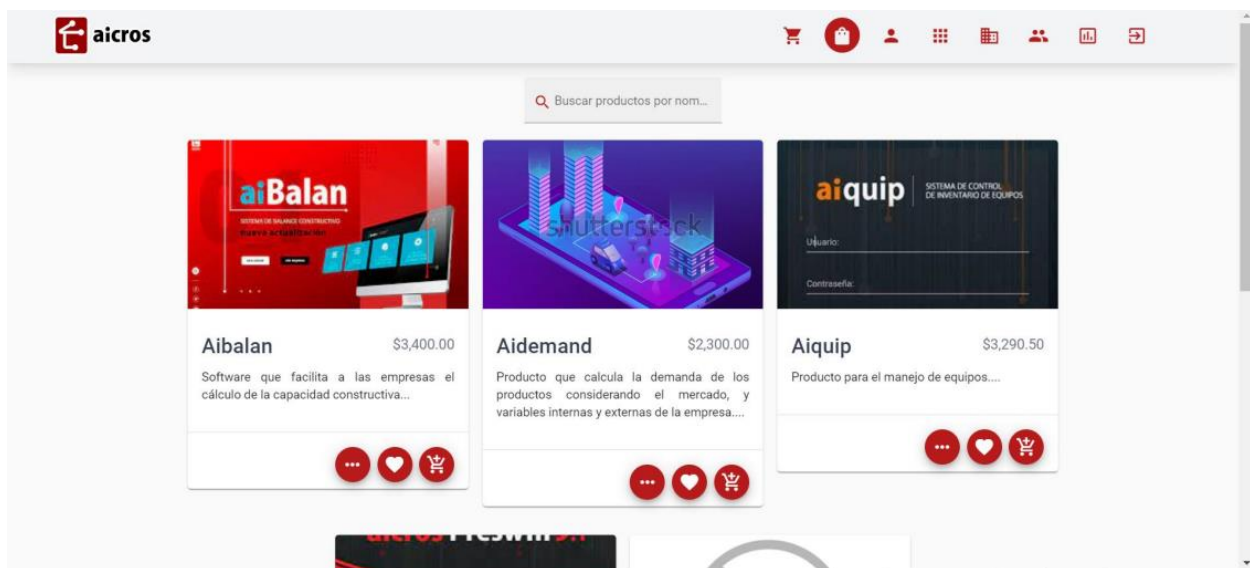


Figura 2.4 Pantalla principal.

Fuente: Elaboración Propia.

Serial	Resolución	Fecha de inicio	Fecha de fin	Representante	Cliente	Tipo Contrato	Archivos	Acciones	
<input checked="" type="checkbox"/>	02/2023	Sergio Martinez Villa	2023-01-01	2027-11-10	Carlos Suarez Naranjo	Roxana Naranjo Benitez	Marco	Archivo 1 Archivo 2	Editar PDF
<input checked="" type="checkbox"/>	01/2023	Res 18/2023	2023-01-07	2025-01-06	Carlos Acosta Paz	GELECT	Prestacion	Archivo 1 Archivo 2	Editar PDF

Items per pagina: 5 1 - 2 de 2

Figura 2.5 Listar contratos. **Fuente:** Elaboración Propia.

Editar Contrato

Resolucion que faculta al representante *
Res24/2022

Fecha inicio
11/16/2022

Representante *
Marcos Lopez Martin

Tipo de contrato *
Marco

Ficha Cliente/Carnet de Identidad Se subió un archivo.

Certifico Legal/Contrato con el banco Se subió un archivo.

Atrás [Guardar](#)

Figura 2.6 Editar contratos. **Fuente:** Elaboración Propia.

Facturado	Serial	Fecha	Total	Subtotal	Acción
FACTURADO	6	09/01/2022	0.00	0.00	Validar
FACTURADO	12	18/01/2022	0.00	0.00	Validar
FACTURADO	11	18/01/2022	0.00	0.00	Validar
FACTURADO	13	20/01/2023	0.00	0.00	Validar
FACTURADO	4	09/01/2023	0.00	0.00	Validar

Figura 2.7 Listar prefacturas. **Fuente:** Elaboración Propia.

A continuación, se muestran las historias de usuarios que se consideran más importante en detalle para que se pueda comprender el proceso:

Tabla 2.7 Diseño y creación de la base de datos.

Historia de usuario	
Número: 1	Usuario: Administrador
Nombre: Diseño y creación de la base de datos	
Prioridad en negocio: Media	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roxana Naranjo Benítez	
Descripción: Se diseña e implementa la base de datos en el gestor <i>PostgreSQL</i> .	
Observaciones: Crear los roles del sistema en el gestor de base de datos para una mayor seguridad.	

Fuente: Elaboración Propia.

Tabla 2.8 Diseño de la interfaz de usuario.

Historia de usuario	
Número: 2	Usuario: Administrador
Nombre: Diseño de la interfaz de usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roxana Naranjo Benítez	
Descripción: Se diseñan las interfaces de la aplicación.	
Observaciones: Resaltar los colores de la empresa y potenciar la experiencia de usuario.	

Fuente: Elaboración Propia.

Tabla 2.9 Listar contratos por cliente.

Historia de usuario	
Número: 3	Usuario: Cliente
Nombre: Listar contratos por clientes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roxana Naranjo Benítez	
Descripción: Se listan los contratos que posee el cliente con la entidad.	
Observaciones: Deben organizarse los contratos por la fecha de creación siendo el primero el último creado	

Fuente: Elaboración Propia.

Tabla 2.10 Gestionar contratos por cliente.

Historia de usuario	
Número: 4	Usuario: Cliente
Nombre: Gestionar contratos por clientes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roxana Naranjo Benítez	
Descripción: El cliente podrá crear un contrato con la entidad y editar los datos de contratos existentes.	
Observaciones: No se podrán editar los contratos marcados como válido	

Fuente: Elaboración Propia.

Tabla 2.11 Listar contratos por jurídico.

Historia de usuario	
Número: 5	Usuario: Jurídico
Nombre: Listar contratos por jurídico	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roxana Naranjo Benítez	
Descripción: Se listan todos los contratos que posee la entidad.	
Observaciones: Deben organizarse los contratos por la fecha de creación siendo el primero el último creado	

Fuente: Elaboración Propia.

Tabla 2.12 Gestionar contratos por jurídico.

Historia de usuario	
Número: 6	Usuario: Jurídico
Nombre: Gestionar contratos por jurídico	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roxana Naranjo Benítez	
Descripción: El jurídico podrá crear contratos a clientes y editarlos según sea necesario.	
Observaciones: No pueden editarse los contratos marcados como válidos	

Fuente: Elaboración Propia.

Tabla 2.13 Gestionar contrato.

Historia de usuario	
Número: 7	Usuario: Cliente
Nombre: Generar contrato	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Roxana Naranjo Benítez	
Descripción: Inicia cuando el usuario rellena el formulario con los datos necesarios para el contrato.	
Observaciones: Se exportará en formato PDF	

Fuente: Elaboración Propia.

Tabla 2.14 Listar prefactura.

Historia de usuario	
Número: 8	Usuario: Comercial
Nombre: Listar prefacturas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roxana Naranjo Benítez	
Descripción: Se listan las prefacturas generadas en la tienda.	
Observaciones: Deben poder filtrarse por el serial	

Fuente: Elaboración Propia

Tabla 2.15 Validar prefactura.

Historia de usuario	
Número: 9	Usuario: Comercial
Nombre: Validar prefacturas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roxana Naranjo Benítez	
Descripción: Se validan las prefacturas una vez que se haya realizado el importe.	
Observaciones: No podrán validarse las prefacturas si no está validado el contrato	

Fuente: Elaboración Propia.

Tabla 2.16 Generar reportes.

Historia de usuario	
Número: 10	Usuario: Administrador
Nombre: Generar reportes	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Roxana Naranjo Benítez	
Descripción: Inicia cuando el usuario accede a ver reportes. Los reportes se mostrarán en gráficas.	
Observaciones: La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación. Si es un jurídico solo podrá ver los reportes de contratos y si es comercial de las facturas	

Fuente: Elaboración Propia.

2.5.2 Tareas de Ingeniería

Es importante descomponer cada una de las historias de usuario en tareas de ingeniería.

Tabla 2.17 Tareas de Ingeniería.

No	Nombre HU	No	Tarea de Ingeniería
1	Diseño y creación de la base de datos	1	Diseñar la base de datos
		2	Crear la base de datos
2	Diseño de la interfaz de usuario	3	Diseñar la interfaz de usuario
3	Listar contratos por cliente	4	Listar contratos cliente
		5	Filtrar contratos
4	Gestionar contratos por cliente	6	Crear contrato
		7	Editar contrato
5	Listar contratos por jurídico	8	Listar contratos jurídico
		9	Filtrar contratos
6	Gestionar contratos por jurídico	10	Crear contrato
		11	Editar contrato
		12	Validar contrato
7	Generar contrato	13	Generar contrato
8	Listar prefacturas	14	Listar prefacturas
		15	Filtrar prefacturas
9	Validar prefacturas	16	Validar prefacturas
10	Generar reportes	17	Reporte de prefacturas por estado
		18	Reporte de ingresos por mes
		20	Reporte contrato por estado

Fuente: Elaboración Propia.

En las tablas que se muestran a continuación se relacionan algunas tareas de ingeniería que tenían mayor peso en el desarrollo de esta investigación.

Tabla 2.16 Diseño de la base de datos.

Tarea de Ingeniería	
Número de tarea: 1	Número de historia de usuario: 1
Nombre: Diseño de la base de datos	
Tipo de tarea: Diseño	Puntos estimados: 0.5
Programador responsable: Roxana Naranjo Benítez	
Descripción: Analizar profundamente el negocio para crear la base de datos que permita almacenar dicha información.	

Fuente: Elaboración Propia.

Tabla 2.19 Creación de la base de datos.

Tarea de Ingeniería	
Número de tarea: 2	Número de historia de usuario: 1
Nombre: Creación de la base de datos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Programador responsable: Roxana Naranjo Benítez	
Descripción: Crear la base de datos con sus respectivas relaciones y la integridad correspondiente entre las tablas.	

Fuente: Elaboración Propia.

Tabla 2.20 Listar contrato jurídico.

Tarea de Ingeniería	
Número de tarea: 8	Número de historia de usuario: 5
Nombre: Listar contrato jurídico	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Roxana Naranjo Benítez	
Descripción: Se mostrará una tabla donde estarán listados todos los contratos realizados con la entidad.	

Fuente: Elaboración Propia.

Tabla 2.21 Crear contrato.

Tarea de Ingeniería	
Número de tarea: 10	Número de historia de usuario: 6
Nombre: Crear contrato	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Roxana Naranjo Benítez	
Descripción: Se mostrará un formulario donde el usuario podrá insertar un contrato con su serial correspondiente.	

Fuente: Elaboración Propia.

Tabla 2.22 Generar contrato.

Tarea de Ingeniería	
Número de tarea: 13	Número de historia de usuario: 7
Nombre: Generar contrato	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Programador responsable: Roxana Naranjo Benítez	
Descripción: Una vez que el usuario halla rellenado los datos en el formulario correspondiente procede a generar el contrato, el cual se le muestra con su estructura final y se da la opción de descargarla en formato PDF.	

Fuente: Elaboración Propia

Tabla 2.23 Reporte de prefacturas por estado

Tarea de Ingeniería	
Número de tarea: 17	Número de historia de usuario: 10
Nombre: Reporte de prefacturas por estado	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Roxana Naranjo Benítez	
Descripción: Se mostrará en un gráfico las prefacturas separadas según el estado en que se encuentran.	

Fuente: Elaboración Propia

2.5.3 Modelo Físico de la Base de Datos

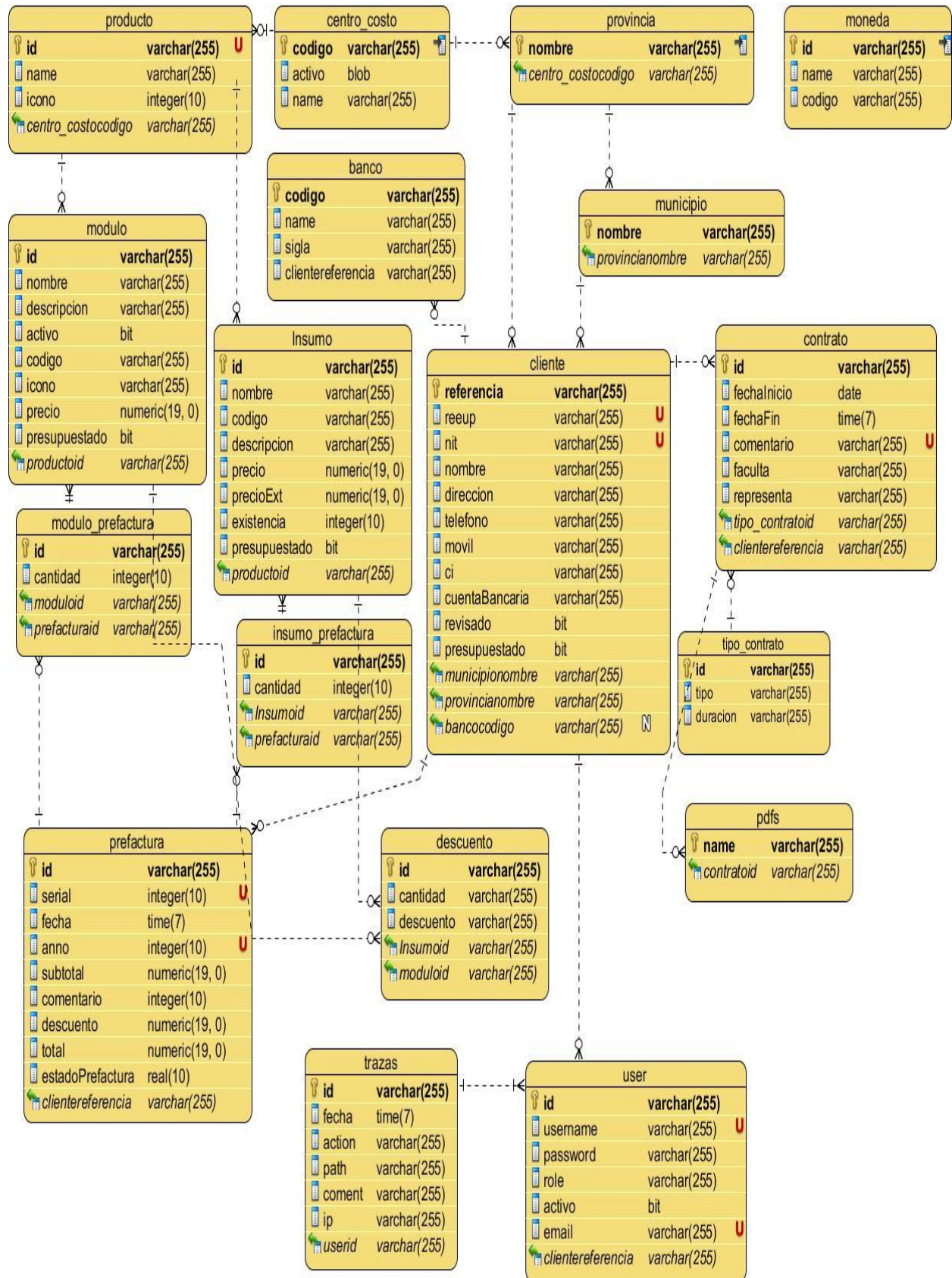


Figura 2.8 Modelo de la base de datos.

Fuente: Elaboración Propia.

Conclusiones parciales del capítulo

Con la culminación de este capítulo queda plasmada claramente la propuesta de la aplicación a implementar según la metodología de desarrollo XP. Se planteó la composición del equipo de desarrollo, las diferentes historias de usuarios y las respectivas interacciones necesarios para acometer la labor de construcción de la aplicación. Se llegó además a la conclusión de que el proyecto es factible, debido a que el costo de producción no es muy elevado y reporta beneficios en el campo de acción seleccionado.

Capítulo III, Elementos de prueba y Resultados obtenidos

Introducción

El siguiente capítulo aborda la descripción de los casos de prueba para los casos de uso definidos con anterioridad, donde se especifican los resultados que debe proporcionar el sistema según la entrada de datos que se haga. Estos son diseñados para disminuir la cantidad de errores a la hora de entrar los datos al sistema y para mantener la mayor integridad posible de los mismos.

3.1 Elementos de prueba al *software*.

El proceso de pruebas es el instrumento más adecuado para determinar el estatus de la calidad de un producto. En este proceso se ejecutan pruebas dirigidas a componentes del *software* o al sistema en su totalidad, con el objetivo de medir el grado en que el este cumple con los requerimientos o si es el que se quería desarrollar. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de Prueba.

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces. Los niveles de trabajo en los cuales se pueden realizar las pruebas son:

- Prueba de Unidad.
- Prueba de Integración.
- Prueba de Sistema.
- Prueba de Aceptación.
- Prueba de Seguridad.

3.2 Plan de pruebas

El plan de pruebas de *software* se elabora con el fin de especificar qué elementos o componentes se van a probar para que el grupo de trabajo pueda realizar el proceso de Validación y Verificación de los requerimientos funcionales y no funcionales. Además, a través del plan de pruebas se puede continuar con la trazabilidad de los requerimientos, con lo cual el grupo de trabajo, identifica el porcentaje de avance que se ha logrado hasta cierto momento (Reyes Chirino & Delgado Fernández, 2021).

Al desarrollar el plan de pruebas, se puede obtener información sobre los errores, defectos o fallas que tiene el prototipo, así se realizan las correcciones pertinentes, según el caso y se asegura la calidad del producto que se está entregando al cliente.

A continuación, se muestra en la Tabla 3.1 el plan de prueba.

Tabla 3.1 Plan de pruebas.

No	Nombre HU	Pruebas a realizar
1	Diseño y creación de la base de datos	Test base de datos
2	Diseño de la interfaz de usuario	Test diseño de la interfaz de usuario
3	Listar contratos por cliente	Test listar contrato por cliente
		Test filtrar contratos por cliente
4	Gestionar contratos por cliente	Test crear contrato por cliente
		Test editar contrato por cliente
5	Listar contratos por jurídico	Test listar contrato por jurídico
		Test filtrar contrato por jurídico
6	Gestionar contratos por jurídico	Test crear contrato por jurídico
		Test editar contrato por jurídico
		Test validar contrato
7	Generar contrato	Test generar contrato
8	Listar prefacturas	Test listar prefacturas
		Test filtrar prefactura
9	Validar prefacturas	Test validar prefacturas
10	Generar reportes	Test generar reportes

Fuente: Elaboración Propia.

3.2.1 Pruebas de aceptación

Cuando se construye *software* a la medida para un cliente, se realiza una serie de pruebas de aceptación a fin de permitir al cliente validar todos los requerimientos. Realizada por el usuario final en lugar de por los ingenieros de software, una prueba de aceptación puede variar desde una “prueba de conducción” informal hasta una serie de pruebas planificadas y ejecutadas sistemáticamente. De hecho, la prueba de aceptación puede realizarse durante un periodo de semanas o meses, y mediante ella descubrir errores acumulados que con el tiempo puedan degradar el sistema (Pressman, 2010).

Uno de los pilares de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos que se vayan implementando. Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto

pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información.

Tabla 3.2 Test base de Datos.

Caso de prueba	
Número: 1	Historia de usuario: 1
Nombre de caso de pruebas: Test base de Datos	
Descripción: Verifica el funcionamiento de la base de datos.	
Condición de ejecución: Estar conectado a la base de datos.	
Entradas: Valores para leer o escribir en la base de datos, por ejemplo: usuario, contraseña.	
Resultado esperado: Se muestran o guardan los datos correctamente.	
Evaluación: Prueba satisfactoria.	

Fuente: Elaboración Propia.

Tabla 3.3 Test gestionar contrato por cliente.

Caso de prueba	
Número: 4	Historia de usuario: 4
Nombre de caso de pruebas: Test gestionar contrato por cliente	
Descripción: Se inserta los datos para crear un nuevo contrato, se inserta de forma incorrecta dejando el campo en blanco, luego se inserta de forma correcta para verificar que los datos sean almacenados. Se modifican los datos de un contrato, se modifican incorrectamente dejando campos en blanco para verificar la validación, se modifican correctamente para que los sean almacenados.	
Condición de ejecución: La aplicación web debe estar en ejecución.	
Entradas: Fecha de inicio del contrato, Resolución que faculta al representante, Representante, tipo de contrato y dos archivos pdf	
Resultado esperado: El sistema notifica al usuario cuando se inserten datos erróneos o cuando se dejen campos en blanco. Cuando se inserten los datos correctamente, se creará un contrato.	
Evaluación: Prueba satisfactoria.	

Fuente: Elaboración Propia

Tabla 3.4 Test gestionar contrato por jurídico.

Caso de prueba	
Número: 6	Historia de usuario: 6
Nombre de caso de pruebas: Test gestionar contrato por jurídico	
Descripción: Se inserta los datos para crear un nuevo contrato, se inserta de forma incorrecta dejando el campo en blanco, luego se inserta de forma correcta para verificar que los datos sean almacenados. Se modifican los datos de un contrato, se modifican incorrectamente dejando campos en blanco para verificar la validación, se modifican correctamente para que los sean almacenados.	
Condición de ejecución: La aplicación web debe estar en ejecución.	
Entradas: Serial, Cliente, Fecha de inicio del contrato, Resolución que faculta al representante, Representante, tipo de contrato y dos archivos pdf	
Resultado esperado: El sistema notifica al usuario cuando se inserten datos erróneos o cuando se dejen campos en blanco. Cuando se inserten los datos correctamente, se creará un contrato.	
Evaluación: Prueba satisfactoria.	

Fuente: Elaboración Propia.

En la Tabla 3.5 y 3.6 se muestran las clases de equivalencia de las pruebas de aceptación al sistema en la HU

Tabla 3.5 Clases de equivalencia

Condición de entrada	Clases válidas	Representante	Clases inválidas	Representante
Fecha Inicio	1. Una fecha que cumpla con el formato dd-mm-aa.	9/7/2022	2. Cadena vacía	NULL
			3. Formato de fecha no correspondiente	8/7/2022
			4. Días que no existen	30-02-2022
Resolución que faculta al representante	5. Cualquier cadena de caracteres alfabéticos. Permite espacio.	Res 03-2022	6. Asteriscos	Res*05-2022
			7. Cadena vacía	NULL
Representante	8. Cualquier cadena de caracteres alfabéticos. Permite espacio.	Ramón Fuentes Lincheta	9. Cadena vacía	NULL
Tipo de contrato	10. Seleccionar alguna de las opciones	Marco	11. No seleccionar ninguna opción	NULL
Ficha Cliente/Carnet de Identidad	12. Subir un archivo pdf	1641790215228\$3962ce13.pdf	13. Subir un archivo que no sea pdf	1662672060803.bd95f0e5.jpg
			14. No subir ningún archivo	NULL
Certifico Legal/Contrato con el banco	15. Subir un archivo pdf	1641790215228\$3962ce13.pdf	16. Subir un archivo que no sea pdf	1662672060803.bd95f0e5.jpg
			17. No subir ningún archivo	NULL

Fuente: Elaboración Propia.

Tabla 3.6 Clases de equivalencia

Condición de entrada	Clases válidas	Representante	Clases inválidas	Representante
Fecha Inicio	1. Una fecha que cumpla con el formato dd-mm-aa.	9/7/2022	2. Cadena vacía	NULL
			3. Formato de fecha no correspondiente	8/7/2022
			4. Días que no existen	30-02-2022
Resolución que faculta al representante	5. Cualquier cadena de caracteres alfabéticos. Permite espacio.	Res 03-2022	6. Asteriscos	Res*05-2022
Representante	8. Cualquier cadena de caracteres alfabéticos. Permite espacio.	Ramón Fuentes Lincheta	7. Cadena vacía	NULL
Tipo de contrato	10. Seleccionar alguna de las opciones	Marco	11. No seleccionar ninguna opción	NULL
			9. Cadena vacía	NULL
Ficha Cliente/Car net de Identidad	12. Subir un archivo pdf	1641790215 228\$3962ce 13.pdf	13. Subir un archivo que no sea pdf	1662672060803.bd95f0e5.jpg
			14. No subir ningún archivo	NULL
Certifico Legal/Contrato con el banco	15. Subir un archivo pdf	1641790215 228\$3962ce 13.pdf	16. Subir un archivo que no sea pdf	1662672060803.bd95f0e5.jpg
			17. No subir ningún archivo	NULL
Serial	18. Cualquier cadena de caracteres alfabéticos. Permite espacio.	Mar09/2022	19. Asteriscos	Mar*05/2022
			20. Cadena vacía	NULL

Fuente: Elaboración Propia.

En la Tabla 3.7 se muestra un resumen de algunos casos de pruebas necesarios para realizar las pruebas de aceptación.

Tabla 3.7 Resumen de Casos de prueba.

No .	Clases equiv.	Fecha inicio	Resolución	Representante	Tipo de contrato	Ficha Cliente/Carnet de Identidad	Certifico Legal/Contrato con el banco	Resultado
1	1,5,8,1 0,12,1 5	7/3/20 22	Res01/ 2022	Juan López Villa	Marco	16417902152 28\$3962ce13 .pdf	6758790215 228\$3962ce 13.pdf	Contrato insertado
2	1,5,8,1 0,12,1 7	7/4/20 22	Res01/ 2023	Juan López Villa	Marco	16417902152 28\$3962ce13 .pdf	NULL	El certfico legal es requerido.
3	1,5,8,1 0,12,1 6	7/5/20 22	Res01/ 2022	Juan López Villa	Marco	16417902152 28\$3962ce13 .pdf	6758790215 228\$3962ce 13.jpg	Debe ser un archivo pdf
4	1,5,8,1 0,13,1 5	7/6/20 22	Res01/ 2022	Juan López Villa	Marco	Null	6758790215 228\$3962ce 13.pdf	La ficha de cliente es requerida
5	1,5,8,1 0,14,1 5	7/7/20 22	Res01/ 2022	Juan López Villa	Marco	67587902152 28\$3962ce13 .jpg	6758790215 228\$3962ce 13.pdf	Debe ser un archivo pdf.
6	1,5,8,1 1,12,1 5	7/8/20 22	Res01/ 2022	Juan López Villa	NULL	67587902152 28\$3962ce13 .pdf	6758790215 228\$3962ce 13.pdf	El tipo de contrato es requerido.
7	1,5,9,1 0,12,1 6	7/9/20 22	Res01/ 2022	NULL	Marco	67587902152 28\$3962ce13 .pdf	6758790215 228\$3962ce 13.pdf	El representante es requerido.
8	1,7,9,1 0,12,1 7	7/10/2 022	NULL	Juan López Villa	Marco	67587902152 28\$3962ce13 .pdf	6758790215 228\$3962ce 13.pdf	La resolución es requerida.
9	1,6,9,1 0,12,1 8	7/11/2 022	Res*01 /2022	Juan López Villa	Marco	67587902152 28\$3962ce13 .pdf	6758790215 228\$3962ce 13.pdf	La resolución debe tener formato Res00/aaaa
10	4,5,9,1 0,12,1 9	31/02/ 2022	Res01/ 2022	Juan López Villa	Marco	67587902152 28\$3962ce13 .pdf	6758790215 228\$3962ce 13.pdf	La fecha es incorrecta
11	3,5,9,1 0,12,2 0	4/5/20 22	Res01/ 2023	Juan López Villa	Marco	67587902152 28\$3962ce13 .pdf	6758790215 228\$3962ce 13.pdf	La fecha debe tener formato dd/mm/aaaa
12	3,5,9,1 0,12,2 1	NULL	Res01/ 2024	Juan López Villa	Marco	67587902152 28\$3962ce13 .pdf	6758790215 228\$3962ce 13.pdf	La fecha es requerida.

Fuente: Elaboración Propia.

A continuación, se muestran algunas Tablas de Prueba por Caso de Prueba

Tabla 3.8 Test insertar contrato caso prueba 1.

Tabla de prueba	
No. Caso de Prueba	1
Requerimiento	La aplicación web debe estar en ejecución
Objetivo	Probar la acción de insertar un contrato (Para cubrir las clases válidas 1,5,8,10,12,15)
Tipo de Prueba	Funcional
Hardware	Sistema de cómputo con un Procesador AMD A9- Disco sólido de 118GB - Memoria RAM de 12GB
Software	Navegador de internet Google Chrome v107.0
Personal	Encargados de pruebas
Datos de entrada	Nombre: Fecha de inicio: 07/05/2022, Resolución: Res02/2022, Representante: Juan López Villa, tipo de contrato: Marco, ficha de cliente: 6758790215228\$3962ce13.pdf, certificado legal: 1324390215228\$3962ce13.pdf
Resultado esperado	Contrato insertado correctamente
Resultados obtenidos	SI (X) NO ()
Casos de excepción	
Aprobado por: Mayli Estopiñán Lantigua	Cargo: Jefe de Proyecto, Tester

Fuente: Elaboración Propia.

Tabla 3.9 Test insertar contrato caso prueba 2.

Tabla de prueba	
No. Caso de Prueba	2
Requerimiento	La aplicación web debe estar en ejecución
Objetivo	Probar la acción de insertar un contrato (Para cubrir las clases válidas 1,5,8,10,12,17)
Tipo de Prueba	Funcional
Hardware	Sistema de cómputo con un Procesador AMD A9- Disco sólido de 118GB - Memoria RAM de 12GB
Software	Navegador de internet Google Chrome v107.0
Personal	Encargados de pruebas
Datos de entrada	Nombre: Fecha de inicio: 07/05/2022, Resolución: Res02/2022, Representante: Juan López Villa, tipo de contrato: Marco, ficha de cliente: 6758790215228\$3962ce13.pdf, certificado legal: NULL
Resultado esperado	El certificado legal es requerido.
Resultados obtenidos	SI(X) NO()
Casos de excepción	
Aprobado por: Mayli Estopiñán Lantigua	Cargo: Jefe de Proyecto, Tester

Fuente: Elaboración Propia.

Tabla 3.10 Test insertar contrato caso prueba 3.

Tabla de prueba	
No. Caso de Prueba	3
Requerimiento	La aplicación web debe estar en ejecución
Objetivo	Probar la acción de insertar un contrato (Para cubrir las clases válidas 1,5,8,10,12,16)
Tipo de Prueba	Funcional
Hardware	Sistema de cómputo con un Procesador AMD A9- Disco sólido de 118GB - Memoria RAM de 12GB
Software	Navegador de internet Google Chrome v107.0
Personal	Encargados de pruebas
Datos de entrada	Nombre: Fecha de inicio: 07/05/2022, Resolución: Res02/2022, Representante: Juan López Villa, tipo de contrato: Marco, ficha de cliente: 6758790215228\$3962ce13.pdf, certificado legal: 6758790215228\$3962ce13.jpg
Resultado esperado	Debe ser un archivo pdf.
Resultados obtenidos	SI(X) NO()
Casos de excepción	
Aprobado por: Mayli Estopiñán Lantigua	Cargo: Jefe de Proyecto, Tester

Fuente: Elaboración Propia.

Tabla 3.11 Test insertar contrato caso prueba 4.

Tabla de prueba	
No. Caso de Prueba	4
Requerimiento	La aplicación web debe estar en ejecución
Objetivo	Probar la acción de insertar un contrato (Para cubrir las clases válidas 1,5,8,10,13,15)
Tipo de Prueba	Funcional
Hardware	Sistema de cómputo con un Procesador AMD A9- Disco sólido de 118GB - Memoria RAM de 12GB
Software	Navegador de internet Google Chrome v107.0
Personal	Encargados de pruebas
Datos de entrada	Nombre: Fecha de inicio: 07/05/2022, Resolución: Res02/2022, Representante: Juan López Villa, tipo de contrato: Marco, ficha de cliente: NULL, certificado legal: 6758790215228\$3962ce13.pdf
Resultado esperado	La ficha de cliente es requerida.
Resultados obtenidos	SI(X) NO ()
Casos de excepción	
Aprobado por: Mayli Estopiñán Lantigua	Cargo: Jefe de Proyecto, Tester

Fuente: Elaboración Propia.

Tabla 3.12 Test insertar contrato caso prueba 5.

Tabla de prueba	
No. Caso de Prueba	5
Requerimiento	La aplicación web debe estar en ejecución
Objetivo	Probar la acción de insertar un contrato (Para cubrir las clases válidas 1,5,8,10,14,15)
Tipo de Prueba	Funcional
Hardware	Sistema de cómputo con un Procesador AMD A9- Disco sólido de 118GB - Memoria RAM de 12GB
Software	Navegador de internet Google Chrome v107.0
Personal	Encargados de pruebas
Datos de entrada	Nombre: Fecha de inicio: 07/05/2022, Resolución: Res02/2022, Representante: Juan López Villa, tipo de contrato: Marco, ficha de cliente: 6758790215228\$3962ce13.jpg, certificado legal: 6758790215228\$3962ce13.pdf
Resultado esperado	Debe ser un archivo pdf.
Resultados obtenidos	SI(X) NO()
Casos de excepción	
Aprobado por: Mayli Estopiñán Lantigua	Cargo: Jefe de Proyecto, Tester

Fuente: Elaboración Propia.

Tabla 3.13 Test insertar contrato caso prueba 6.

Tabla de prueba	
No. Caso de Prueba	6
Requerimiento	La aplicación web debe estar en ejecución
Objetivo	Probar la acción de insertar un contrato (Para cubrir las clases válidas 1,5,8,11,12,15)
Tipo de Prueba	Funcional
Hardware	Sistema de cómputo con un Procesador AMD A9- Disco sólido de 118GB - Memoria RAM de 12GB
Software	Navegador de internet Google Chrome v107.0
Personal	Encargados de pruebas
Datos de entrada	Nombre: Fecha de inicio: 07/05/2022, Resolución: Res02/2022, Representante: Juan López Villa, tipo de contrato: NULL, ficha de cliente: 6758790215228\$3962ce13.pdf, certificado legal: 6758790215228\$3962ce13.pdf
Resultado esperado	El tipo de contrato requerido.
Resultados obtenidos	SI(X) NO()
Casos de excepción	
Aprobado por: Mayli Estopiñán Lantigua	Cargo: Jefe de Proyecto, Tester

Fuente: Elaboración Propia.

Tabla 3.14 Test insertar contrato caso prueba 7.

Tabla de prueba	
No. Caso de Prueba	7
Requerimiento	La aplicación web debe estar en ejecución
Objetivo	Probar la acción de insertar un contrato (Para cubrir las clases válidas 1,5,9,10,12,15)
Tipo de Prueba	Funcional
Hardware	Sistema de cómputo con un Procesador AMD A9- Disco sólido de 118GB - Memoria RAM de 12GB
Software	Navegador de internet Google Chrome v107.0
Personal	Encargados de pruebas
Datos de entrada	Nombre: Fecha de inicio: 07/05/2022, Resolución: Res02/2022, Representante: NULL, tipo de contrato: Marco, ficha de cliente: 6758790215228\$3962ce13.pdf, certificado legal: 6758790215228\$3962ce13.pdf
Resultado esperado	El representante requerido.
Resultados obtenidos	SI(X) NO()
Casos de excepción	
Aprobado por: Mayli Estopiñán Lantigua	Cargo: Jefe de Proyecto, Tester

Fuente: Elaboración Propia.

Tabla 3.15 Test insertar contrato caso prueba 8.

Tabla de prueba	
No. Caso de Prueba	8
Requerimiento	La aplicación web debe estar en ejecución
Objetivo	Probar la acción de insertar un contrato (Para cubrir las clases válidas 1,7,9,10,12,15)
Tipo de Prueba	Funcional
Hardware	Sistema de cómputo con un Procesador AMD A9- Disco sólido de 118GB - Memoria RAM de 12GB
Software	Navegador de internet Google Chrome v107.0
Personal	Encargados de pruebas
Datos de entrada	Nombre: Fecha de inicio: 07/05/2022, Resolución: NULL, Representante: Juan López Villa, tipo de contrato: Marco, ficha de cliente: 6758790215228\$3962ce13.pdf, certificado legal: 6758790215228\$3962ce13.pdf
Resultado esperado	La resolución es requerida.
Resultados obtenidos	SI(X) NO ()
Casos de excepción	
Aprobado por: Mayli Estopiñán Lantigua	Cargo: Jefe de Proyecto, Tester

Fuente: Elaboración Propia.

Tabla 3.16 Test insertar contrato caso prueba 9.

Tabla de prueba	
No. Caso de Prueba	9
Requerimiento	La aplicación web debe estar en ejecución
Objetivo	Probar la acción de insertar un contrato (Para cubrir las clases válidas 1,6,9,10,12,15)
Tipo de Prueba	Funcional
Hardware	Sistema de cómputo con un Procesador AMD A9- Disco sólido de 118GB - Memoria RAM de 12GB
Software	Navegador de internet Google Chrome v107.0
Personal	Encargados de pruebas
Datos de entrada	Nombre: Fecha de inicio: 07/05/2022, Resolución: Res*01/2022, Representante: Juan López Villa, tipo de contrato: Marco, ficha de cliente: 6758790215228\$3962ce13.pdf, certificado legal: 6758790215228\$3962ce13.pdf
Resultado esperado	La resolución debe tener formato Res00/aaaa.
Resultados obtenidos	SI(X) NO ()
Casos de excepción	
Aprobado por: Mayli Estopiñán Lantigua	Cargo: Jefe de Proyecto, Tester

Fuente: Elaboración Propia.

Tabla 3.17 Test insertar contrato caso prueba 19.

Tabla de prueba	
No. Caso de Prueba	10
Requerimiento	La aplicación web debe estar en ejecución
Objetivo	Probar la acción de insertar un contrato (Para cubrir las clases válidas 4,5,9,10,12,15)
Tipo de Prueba	Funcional
Hardware	Sistema de cómputo con un Procesador AMD A9- Disco sólido de 118GB - Memoria RAM de 12GB
Software	Navegador de internet Google Chrome v107.0
Personal	Encargados de pruebas
Datos de entrada	Nombre: Fecha de inicio: 31/02/2022, Resolución: Res02/2022, Representante: Juan López Villa, tipo de contrato: Marco, ficha de cliente: 6758790215228\$3962ce13.pdf, certificado legal: 6758790215228\$3962ce13.pdf
Resultado esperado	La fecha es incorrecta.
Resultados obtenidos	SI(X) NO ()
Casos de excepción	
Aprobado por: Mayli Estopiñán Lantigua	Cargo: Jefe de Proyecto, Tester

Fuente: Elaboración Propia.

Tabla 3.18 Test insertar contrato caso prueba 11.

Tabla de prueba	
No. Caso de Prueba	11
Requerimiento	La aplicación web debe estar en ejecución
Objetivo	Probar la acción de insertar un contrato (Para cubrir las clases válidas 3,5,9,10,12,15)
Tipo de Prueba	Funcional
Hardware	Sistema de cómputo con un Procesador AMD A9- Disco sólido de 118GB - Memoria RAM de 12GB
Software	Navegador de internet <i>Google Chrome</i> v107.0
Personal	Encargados de pruebas
Datos de entrada	Nombre: Fecha de inicio: 2022/02/03, Resolución: Res02/2022, Representante: Juan López Villa, tipo de contrato: Marco, ficha de cliente: 6758790215228\$3962ce13.pdf, certificado legal: 6758790215228\$3962ce13.pdf
Resultado esperado	La fecha debe tener formato mm/dd/aaaa.
Resultados obtenidos	SI(X) NO()
Casos de excepción	
Aprobado por: Mayli Estopiñán Lantigua	Cargo: Jefe de Proyecto, Tester

Fuente: Elaboración Propia.

Tabla 3.19 Test insertar contrato caso prueba 12.

Tabla de prueba	
No. Caso de Prueba	12
Requerimiento	La aplicación web debe estar en ejecución
Objetivo	Probar la acción de insertar un contrato (Para cubrir las clases válidas 2,5,9,10,12,15)
Tipo de Prueba	Funcional
Hardware	Sistema de cómputo con un Procesador AMD A9- Disco sólido de 118GB - Memoria RAM de 12GB
Software	Navegador de internet <i>Google Chrome</i> v107.0
Personal	Encargados de pruebas
Datos de entrada	Nombre: Fecha de inicio: NULL, Resolución: Res02/2022, Representante: Juan López Villa, tipo de contrato: Marco, ficha de cliente: 6758790215228\$3962ce13.pdf, certificado legal: 6758790215228\$3962ce13.pdf
Resultado esperado	La fecha es requerida.
Resultados obtenidos	SI(X) NO ()
Casos de excepción	
Aprobado por: Mayli Estopiñán Lantigua	Cargo: Jefe de Proyecto, Tester

Fuente: Elaboración Propia.

Al *software* también se le realizaron pruebas de seguridad con el software Vega.

Scan Alert Summary

High		(9 found)
Insecure Cross-Origin Resource Access Control	7	
Possible Social Security Number Detected	1	
Possible Social Insurance Number Detected	1	
Medium		(1 found)
Local Filesystem Paths Found	1	
Low		(None found)
Info		(7 found)
X-Frame-Options Header Not Set	7	

Figura 3.1 Prueba de seguridad. **Fuente:** Elaboración Propia.

Se midió la calidad de la experiencia del usuario con *Lighthouse*.

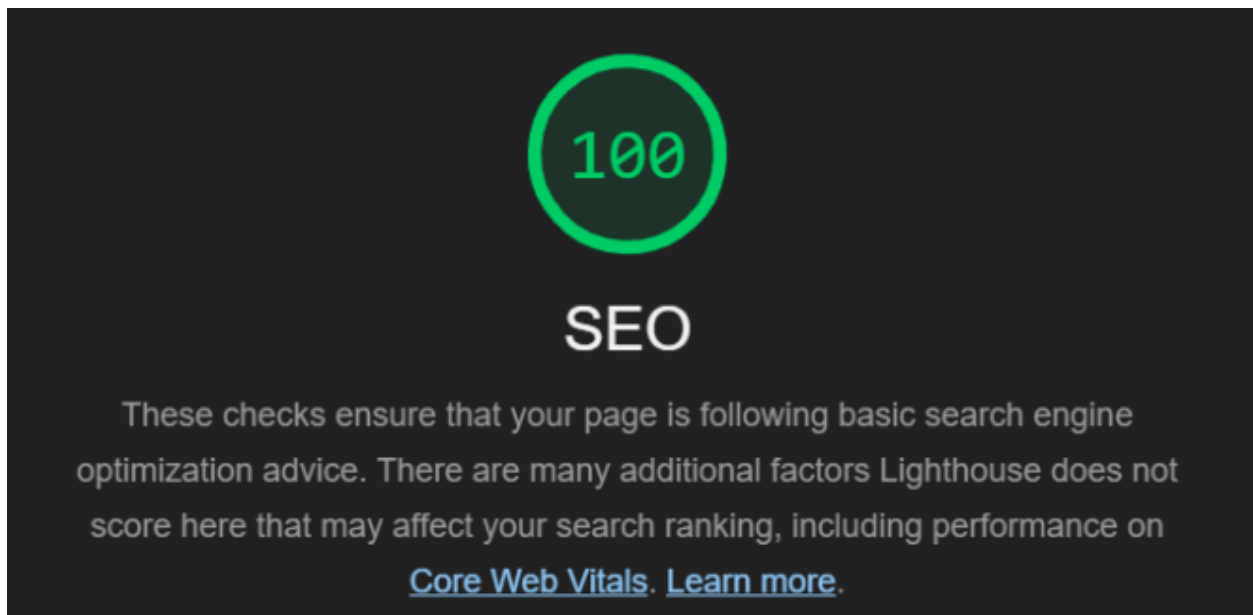


Figura 3.2 Prueba de seguridad. **Fuente:** Elaboración Propia.

3.2 Análisis de los resultados obtenidos.

Con la realización de las pruebas correspondientes se presentaron algunas situaciones en las que el usuario pudiera cometer errores y se analizó en cada caso cómo debe responder el sistema en aras de disminuir o eliminar las probabilidades de errores en la inserción de los datos. Al ejecutar estos casos de prueba se verificó que el sistema funciona correctamente según los objetivos e hipótesis planteados. Esta etapa es de vital importancia porque sirve de comprobación de la eficacia de la aplicación y su nivel de respuesta ante la situación problemática planteada inicialmente.

Conclusiones

Con la realización de los casos de prueba planteados anteriormente se ha podido constatar la efectividad del sistema para solucionar la situación problemática trazada en la investigación, quedó demostrado que el sistema funciona correctamente y que está listo para utilizarse en el 100% de las opciones que brinda.

Conclusiones

Luego de haber realizado un estudio exhaustivo de la problemática abordada en la actual investigación y lográndose dar respuesta a ella de manera satisfactoria, se pudo arribar a las siguientes conclusiones:

1. El estudio realizado sobre los antecedentes y documentos relacionados con el objeto de estudio permitió confeccionar un marco teórico referencial que proporcionó los elementos necesarios para dar solución a la problemática abordada.
2. Se evidenció que las herramientas y tecnologías seleccionadas son las más adecuadas para las características del sistema a elaborar.
3. Se desarrolló un módulo para la gestión de contratos y facturas en la empresa que satisface las necesidades del cliente.
4. Mediante la aplicación de pruebas a la aplicación Web, resultó posible obtener resultados favorables, con el consiguiente análisis de errores detectados que fueron subsanados como parte del desarrollo de esta aplicación.

De manera general se concluye que el proyecto es factible, contribuye a la eficiencia del proceso y reporta beneficios en el campo de acción seleccionado.

Recomendaciones

En función de lograr en un futuro una tienda virtual multifuncional, recomendamos:

1. Incluir la firma electrónica para mejorar aún más el proceso de contratación.
2. Continuar la ampliación de la aplicación mediante la incorporación de nuevas funcionalidades que brinden un mayor control y análisis de los datos almacenados.
3. Proporcionar un número superior de notificaciones, reportes y gráficos que posibiliten una interpretación superior del flujo de información.

Bibliografía

- ¿Qué es Angular? Características y ventajas. (20 de junio de 2022). Recuperado el 11 de noviembre de 2022, de HobSpot: blog.hubspot.es
- Álvarez, M. A. (2020). *Manual de Nestjs*.
- Álvarez, S. (2017). *Sistemas gestores de bases de datos*.
- Calvo, M. (23 de septiembre de 2021). *La importancia de la gestión digital de facturas en una empresa*. Recuperado el 22 de noviembre de 2022, de Contabilidad y fiscalía: www.captio.net/blog/importancia-gestion-digital-empresa
- Comisión Nacional de los mercados y la competencia (CNMC). (2016). Recuperado el 22 de septiembre de 2022, de <https://docs.google.com/document/d/1n65Mj/>
- Cristancho, F. (25 de julio de 2022). ¿Qué es Angular? (Talently) Recuperado el 8 de noviembre de 2022, de talently.tech: www.talently.tech.com
- Cuena, M. (2020). La contratación a través de plataformas intermediarias en línea. *12*(2).
- Desarrollo web. (2021). *HTML*. Recuperado el 21 de noviembre de 2022, de Desarrollo web.
- Desarrollo web. (s.f.). *Manual de CSS*. Recuperado el 20 de noviembre de 2022, de Desarrollo web.
- desarrolloweb. (21 de septiembre de 2021). *Desarrollo Web*. (EscuelaIT) Recuperado el 06 de junio de 2022, de Introduccion a Nest.Js: www.desarrolloweb.com
- EGA Futura. (2022). *Informe de estadísticas de comercio 2022 de Groove*. Recuperado el 05 de 10 de 2022, de EGA Futura: <https://discover.egafutura.com/informe-de-estadisticas-de-comercio-electronico-de-2022-de-groove/>
- Flores, F. (2022). *Qué es Visual Studio Code y qué ventajas ofrece*. Recuperado el 25 de noviembre de 2022, de OpenWebinars: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- Gestión de contratos*. (26 de septiembre de 2022). Recuperado el 2016 de noviembre de 2022, de tic.portal: www.ticportal.es
- Gil, A. (17 de febrero de 2022). *Evolución de la factura electrónica en América Latina*. Recuperado el 22 de noviembre de 2022, de Seres Blog LATAM.
- Git. (2022). *Git*. Recuperado el 25 de noviembre de 2022, de Git: git-scm.com
- Grupo de desarrollo global de PostgreSQL. (2022). *Postgre*. Recuperado el 12 de octubre de 2022, de <https://db.cs.berkeley.edu/postgres.html>
- Guerra Velázquez, D. &. ((2013)). *Solución informática para mejorar la calidad de los datos en el Sistema de Gestión Universitaria*. La Habana.: Universidad de las Ciencias Informáticas.
- Hernández Valverde. (2004). *Estudio de la Relación entre Arquitectura de Software y Usabilidad*. DOI.
- Joskowicz., J. (2008). *Reglas y prácticas en Xtreme Programming*. Universidad de Vigo.

- Laínez Fuentes, J. R. (2017). *Desarrollo de Software Ágil. Extreme Programming y Scrum*. IT Campus Academy.
- Martínez, J. A. (2015). Sistema para la gestión de Oportunidades de Negocio para el Centro de Informática Industrial.
- Martinez-Marsal, A., & Guia-Alcolea, K. (2021). Factura digital: un reto para el comercio en Cuba. *Número especial*.
- Microsoft. (2022). *TypeScript*. (Microsoft) Recuperado el 30 de noviembre de 2022, de www.typescript.org
- Microsoft. (2022). *Visual Studio Code*. (Microsoft) Recuperado el 2022, de code.visualstudio.com
- Ministerio de Justicia. (2013). Decreto Ley número 304 de la contratación económica.
- Ministerio de Justicia. (2021). Resolución 55 del 2021.
- Nestjs*. (2022). Recuperado el 12 de julio de 2022, de Desarrollo Web: <https://docs.nestjs.com/>
- NTT DATA. (8 de abril de 2020). *Qué es Node.js y primeros pasos*. Recuperado el 21 de noviembre de 2022, de ifGeekthen.
- ODOO Community. (2022). *ODOO*. (ODOO) Recuperado el 06 de julio de 2022, de ODOO: www.odoo.com
- Pressman, R. S. (2010). *Ingeniería del software: Un enfoque práctico (Séptima edición ed.)*. MCGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.
- Redacción APD. (13 de 01 de 2022). *Como aplicar la metodología Scrum y que es el metodo Scrum*. Recuperado el 30 de 06 de 2022, de APD: www.apd.es
- Sánchez, J. A. (mayo de 2017). *Arquitectura n-capas*. Recuperado el 25 de noviembre de 2022, de Administración de Base de datos: iutll-abdd-blogspot.com/architectura-ncapas
- Sharp, B. L. (2011). *Introducing HTML5*.
- Stackscale.S.L. (26 de 04 de 2022). *Crecimiento y estadísticas del Ecommerce*. Recuperado el 05 de 10 de 2022, de Stackscale: www.stackscale.com
- The PostgreSQL Global Development Group. (2010). *PostgreSQL Documentation*. California.
- Urbon López, R. (2022). RETOS PARA EL DERECHO CONTRACTUAL CUBANO EN CORRESPONDENCIA CON LOS NUEVOS MODELOS CONTRACTUALES EMERGENTES. (16).
- Varela Vaca, J. R. (2018). El Sistema de contratación pública y su impacto en el desarrollo de las MIPYMES del Cantón Santa Elena. La Libertad, Ecuador.
- Visual Paradigm Internacional. (s.f.). *Visual Paradigm*. (Visual Paradigm Internacional) Recuperado el 30 de noviembre de 2022, de Visual Paradigm: www.visual-paradigm.com
- Walsh, B. (2011). Today's Smart Choice: Don't Own.Share.

Anexos

Anexo 1 Tabla de ponderaciones para EI, EQ, EO, ILF, EIF.

Clasificación de entradas y consultas	1 – 4 Atributos	5 – 15 Atributos	Mas de 15 Atributos
0 o 1 ficheros accedidos	BAJA 3	BAJA 3	MEDIA 4
2 ficheros accedidos	BAJA 3	MEDIA 4	ALTA 6
Más de 2 ficheros accedidos	MEDIA 4	ALTA 6	ALTA 6

Clasificación de entradas y consultas	1 – 5 Atributos	5 – 19 Atributos	Mas de 19 Atributos
0 o 1 ficheros accedidos	BAJA 4	BAJA 4	MEDIA 5
2 a 3 ficheros accedidos	BAJA 4	MEDIA 5	ALTA 7
Más de 5 Entidades o registros lógicos	MEDIA 5	ALTA 7	ALTA 7

Clasificación de entradas y consultas	1 – 19 Atributos	20 – 50 Atributos	Mas de 50 Atributos
1 entidad o registro lógico	BAJA 7	BAJA 7	MEDIA 10
2 - 5 Entidades o registros lógicos	BAJA 7	MEDIA 10	ALTA 15
Más de 2 ficheros accedidos	MEDIA 10	ALTA 15	ALTA 15

Clasificación de entradas y consultas	1 – 4 Atributos	5 – 15 Atributos	Mas de 15 Atributos
0 o 1 ficheros accedidos	BAJA 5	BAJA 5	MEDIA 7
2 ficheros accedidos	BAJA 5	MEDIA 7	ALTA 10
Más de 2 ficheros accedidos	MEDIA 7	ALTA 10	ALTA 10

Fuente: Elaboración Propia.