



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 4

Título:

Aplicación Web para el Proceso de Gestión de Contratación y
Facturación de Servicios Informáticos de la UEB-Datos de la Empresa
de Servicios Especializados GET

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

Autor: Idalia González Brito

Tutores: Ing. Alfonso Hernández Ramírez

M.Sc. Yadira Ramírez Rodríguez

La Habana, 2023

“Año 64 de la Revolución”

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título ***“Aplicación Web para el Sistema de Gestión Comercial en el Proceso de Contratación y Facturación de Servicios Informáticos de la UEB-Datos de la Empresa de Servicios Especializados GET”*** concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los 27 días del mes de noviembre del año 2023.

Idalia González Brito



Firma del Autor

Alfonso Hernández Ramírez



Firma del Tutor

Yadira Ramírez Rodríguez



Firma del Tutor

DATOS DE CONTACTO

Yadira Ramírez Rodríguez, graduada de Ingeniería en Ciencias Informáticas en el año 2007. Máster en Calidad de Software y profesora auxiliar con más de 15 de años de experiencia en la asignatura de Ingeniería de Software. Jefa del Departamento Docente de Informática de la Facultad 4 de la Universidad de las Ciencias Informáticas. Líneas de Investigación: Ingeniería de Sistemas y Calidad de Software.

Correo electrónico: yramirezr@uci.cu

Alfonso Hernández Ramírez, graduado de Ingeniería en Ciencias Informáticas en el año 2019, con experiencia en desarrollo de aplicaciones informáticas y administración de redes, Especialista B en Ciencias Informática (Jefe de Brigada de soporte Técnico de Datos) de la Empresa Servicios Especializados GET. Línea de trabajo y/o investigación: Desarrollo de Aplicaciones Web, la Administración de Redes y la Ciberseguridad.

Correo electrónico: alfonso.hernandes@get.tur.cu

AGRADECIMIENTOS

Agradezco a mi esposo Ernesto por ser mi fiel compañero todos estos años, por brindarme su amor incondicional y por ayudarme y apoyarme durante todo el transcurso de la carrera, este logro también es tuyo.

A mis padres y a mis abuelos por su sacrificio, amor inquebrantable y comprensión durante toda mi vida y en especial en mi etapa universitaria. Todos mis logros son para ustedes que han sabido ser ejemplo en todo para mí.

A mi hermana Nancy por estar siempre a mi lado y brindarme su apoyo incondicional.

A los padres de mi esposo, en especial a su mamá por ser como una segunda mamá para mí y siempre estar ahí para nosotros con su energía y mente positiva.

A mis compañeros de trabajo en especial a los del área comercial de la UEB-Dato, así como a mi jefe y tutor Alfonso por el tiempo empleado en guiarme y apoyarme en el desarrollo de este trabajo de investigación.

A mis amigos, compañeros de carrera y profesores que a lo largo de los años me han brindado todo su apoyo y conocimientos, sin ustedes no hubiese sido posible que llegara donde estoy.

A mis tutores por su esfuerzo y sacrificio para ayudarme a alcanzar esta meta.

¡¡¡A todos les estaré eternamente agradecida!!!

RESUMEN

El auge de las nuevas tecnologías y la aplicación de la informática en el perfeccionamiento empresarial suscitan el desarrollo creciente de la rama tecnológica en el afán de optimizar el rendimiento de las mismas. En el departamento comercial dentro de la Empresa de Servicios Especializados GET en el área UEB Datos, la información que se maneja es de forma manual con la ayuda del tabulador electrónico Excel, donde se torna engorroso el manejo y gestión de la información y en algunos casos podría perderse a partir de algún error humano al introducir los datos o rotura de la computadora. Además, si alguna persona externa a ese departamento necesita consultar datos procesados por los trabajadores del mismo, deben ir hasta la PC que contiene la información, consultar los documentos físicos realizados o pedir un envío de la información digital por correo electrónico u otro medio electrónico. Para dar solución a estos problemas, se desarrolló una aplicación web para el Proceso de Gestión de Contratación y Facturación de Servicios Informáticos de la UEB-Datos de la Empresa de Servicios Especializados GET. La aplicación permite la gestión de los Contratos de Arrendamiento, de Comodato, de Prestación de Servicios y Documentación de Proyectos, así como gestionar los clientes y servicios de la UEB y generar las facturas de los servicios solicitados. Para guiar el proceso de desarrollo del software se utilizó la metodología Scrum; y las herramientas y tecnologías empleadas para la propuesta de solución fueron: HTML, CSS, JavaScript, Vue3, Node.js con Express.js, PostgreSQL, Docker, Git, Github y Postman.

PALABRAS CLAVE: cliente, contrato, factura, servicios.

ABSTRACT

The rise of new technologies and the application of information technology in business improvement lead to the growing development of the technological branch in the desire to optimize their performance. In the commercial department within the GET Specialized Services Company in the UEB Data area, the information that is handled is manually with the help of the Excel electronic tabulator, where the handling and management of the information becomes cumbersome, and in some cases could be lost due to human error when entering data or computer breakdown. Furthermore, if anyone external to that department needs to consult data processed by its workers, they must go to the PC that contains the information, consult the physical documents made or request a sending of the digital information by email or other electronic means. To solve these problems, a web application was developed for the Computer Services Contracting and Billing Management Process of the UEB-Data of the Specialized Services Company GET. The application allows the management of Lease Contracts, Comodato, Provision of Services and Project Documentation, as well as managing the clients and services of the UEB and generating invoices for the requested services. To guide the software development process, the Scrum methodology was used; and the tools and technologies used for the solution proposal were: HTML, CSS, JavaScript, Vue3, Node.js with Express.js, PostgreSQL, Docker, Git, Github and Postman.

KEYWORDS: client, contract, invoice, services.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO I: Fundamentación Teórica	5
1.1 Definición de los principales conceptos relacionados con el objeto de estudio y campo de acción.	5
1.2 Sistemas Homólogos.....	8
1.2.1 Análisis específico de los sistemas homólogos.	11
1.2.2 Valoración crítica de los sistemas estudiados	11
1.3 Ambiente de desarrollo.....	12
1.3.1 Metodología de desarrollo de software.....	12
1.3.2 Caracterización de las tecnologías y las herramientas.....	14
Conclusiones del capítulo.....	22
CAPÍTULO II: Diseño de la propuesta de solución al problema de investigación.	24
2.1 Descripción de la solución	24
2.3 Requisitos de software	25
2.3.1 Requisitos funcionales	25
2.3.2 Requisitos no funcionales	28
2.3.3. Historias de Usuario.....	29
2.4 Planificación del Backlog	33
2.4.1 Product Backlog.....	33
2.4.2 Sprint Backlog.....	37
2.5 Arquitectura de la solución	47
2.5.1 Patrón arquitectónico Modelo – Vista – Controlador (MVC)	47
Conclusiones del capítulo.....	49
CAPÍTULO III: Validación e implementación de la propuesta de solución.....	50
	VI

3.1 Fase de despliegue	50
3.2 Estándares de codificación	51
3.2 Pruebas de software.....	52
3.2.1 Estrategia de prueba.....	52
3.2.2 Niveles de prueba	53
3.2.3 Pruebas unitarias	53
3.2.5 Pruebas de aceptación	54
3.2.6 Pruebas de integración	55
3.2.7 Resultado de las pruebas aplicadas.....	56
Conclusiones del capítulo.....	57
CONCLUSIONES FINALES	58
RECOMENDACIONES.....	59
REFERENCIAS BIBLIOGRÁFICAS	60
ANEXOS.....	66
Anexo 1: Historias de usuario.....	66
Anexo 2: Pruebas de aceptación.....	84

ÍNDICE DE TABLAS

Tabla 1-1 Pros y Contras de Odoo.	9
Tabla 1-2 Pros y Contras de Holden.	10
Tabla 1-3 Análisis específico de los sistemas homólogos.	11
Tabla 1-4 Comparación entre Angular, React y Vue.	16
Tabla 2-1 Requisitos Funcionales.	26
Tabla 2-2 Historia de Usuario: “HU01 - Autenticar usuario”	30
Tabla 2-3 Historia de Usuario: “HU07 - Editar Cliente”	31
Tabla 2-4 Product Backlog.	35
Tabla 2-5 Sprint backlog 1.	39
Tabla 2-6 Sprint backlog 2.	44
Tabla 2-7 Sprint backlog 3.	47
Tabla 3-1 Prueba de Aceptación.	56
Tabla 3-2 No conformidades.	58
Tabla A1-1 Historia de usuario “HU02 - Registrar usuario”	66
Tabla A1-2 Historia de usuario “HU03 - Editar usuario”	66
Tabla A1-3 Historia de usuario “HU04 - Listar usuario”	67
Tabla A1-4 Historia de usuario “HU05 – Deshabilitar Usuario”	67
Tabla A1-5 Historia de usuario “HU06 – Añadir Cliente”	68
Tabla A1-6 Historia de usuario “HU08 – Listar cliente”	69
Tabla A1-7 Historia de usuario “HU09 – Añadir Tipo de Servicio”	69
Tabla A1-8 Historia de usuario “HU10 – Editar Tipo de Servicio”	70
Tabla A1-9 Historia de usuario “HU11 – Listar Tipo de Servicio”	70
Tabla A1-10 Historia de usuario “HU12 – Añadir Servicio”	71
Tabla A1-11 Historia de usuario “HU13 – Editar Servicio”	71
Tabla A1-12 Historia de usuario “HU14 – Listar Servicio”	72
Tabla A1-13 Historia de usuario “HU15 – Añadir Solicitud de Servicio”	72
Tabla A1-14 Historia de usuario “HU16 – Editar Solicitud de Servicio”	73
Tabla A1-15 Historia de usuario “HU17 – Listar Solicitud de Servicio”	74
Tabla A1-16 Historia de usuario “HU18 – Crear Oferta Comercial”	74

Tabla A1-17 Historia de usuario “HU19 – Editar Oferta Comercial”	75
Tabla A1-18 Historia de usuario “HU20 – Listar Oferta Comercial”	76
Tabla A1-19 Historia de usuario “HU21 – Imprimir Oferta Comercial”	76
Tabla A1-20 Historia de usuario “HU22 – Crear Contrato”	77
Tabla A1-21 Historia de usuario “HU23 – Listar Contrato”	78
Tabla A1-22 Historia de usuario “HU24 – Imprimir Contrato”	78
Tabla A1-23 Historia de usuario “HU25 – Crear Suplemento”	78
Tabla A1-24 Historia de usuario “HU26 – Listar Suplemento”	79
Tabla A1-25 Historia de usuario “HU27 – Añadir Reporte de Producción”	80
Tabla A1-26 Historia de usuario “HU28 – Editar Reporte de Producción”	81
Tabla A1-27 Historia de usuario “HU29 – Listar Reporte de Producción”	82
Tabla A1-28 Historia de usuario “HU30 – Imprimir Reporte de Producción”	82
Tabla A1-29 Historia de usuario “HU31 – Imprimir Factura”	83
Tabla A1-30 Historia de usuario “HU32 – Generar el Contrato a imprimir”	83
Tabla A1-31 Historia de usuario “HU33 – Generar la Factura a imprimir”	84
Tabla A2-1 Pruebas de aceptación “PA2 – Registrar usuario”	84
Tabla A2-2 Pruebas de aceptación “PA3 – Editar usuario”	85
Tabla A2-3 Pruebas de aceptación “PA4 – Listar usuario”	87
Tabla A2-4 Pruebas de aceptación “PA5 – Deshabilitar usuario”	87
Tabla A2-5 Pruebas de aceptación “PA6 – Añadir usuario”	88
Tabla A2-6 Pruebas de aceptación “PA7 – Editar Cliente”	88
Tabla A2-7 Pruebas de aceptación “PA8 – Listar cliente”	88
Tabla A2-8 Pruebas de aceptación “PA9 – Añadir Tipo de Servicio”	89
Tabla A2-9 Pruebas de aceptación “PA10 – Editar Tipo de Servicio”	89
Tabla A2-10 Pruebas de aceptación “PA11 – Listar Tipo de Servicio”	90
Tabla A2-11 Pruebas de aceptación “PA12 – Añadir Servicio”	90
Tabla A2-12 Pruebas de aceptación “PA13 – Editar Servicio”	90
Tabla A2-13 Pruebas de aceptación “PA14 – Listar Servicio”	91
Tabla A2-14 Pruebas de aceptación “PA15 – Añadir Solicitud de Servicio”	91
Tabla A2-15 Pruebas de aceptación “PA16 – Editar Solicitud de Servicio”	92
Tabla A2-16 Pruebas de aceptación “PA17 – Listar Solicitud de Servicio”	92
	IX

Tabla A2-17 Pruebas de aceptación “PA18 – Crear Oferta Comercial”	92
Tabla A2-18 Pruebas de aceptación “PA19 – Editar Oferta Comercial”	93
Tabla A2-19 Pruebas de aceptación “PA20 – Listar Oferta Comercial”	93
Tabla A2-20 Pruebas de aceptación “PA21 – Imprimir Oferta Comercial”	94
Tabla A2-21 Pruebas de aceptación “PA22 – Crear Contrato”	94
Tabla A2-22 Pruebas de aceptación “PA23 – Listar Contrato”	95
Tabla A2-23 Pruebas de aceptación “PA24 – Imprimir Contrato”	95
Tabla A2-24 Pruebas de aceptación “PA25 – Crear Suplemento”	95
Tabla A2-25 Pruebas de aceptación “PA26 – Listar Suplemento”	96
Tabla A2-26 Pruebas de aceptación “PA27 – Añadir Reporte de Producción”	96
Tabla A2-27 Pruebas de aceptación “PA28 – Editar Reporte de Producción”	97
Tabla A2-28 Pruebas de aceptación “PA29 – Listar Reporte de Producción”	97
Tabla A2-29 Pruebas de aceptación “PA30 – Imprimir Reporte de Producción”	98
Tabla A2-30 Pruebas de aceptación “PA31 – Imprimir Factura”	98
Tabla A2-31 Pruebas de aceptación “PA32 – Generar el Contrato a imprimir”	98
Tabla A2-32 Pruebas de aceptación “PA33 – Generar la Factura a imprimir”	99

ÍNDICE DE FIGURAS

Figura 2-1 Propuesta de Solución.	25
Figura 2-2 Modelo de la arquitectura del lado del servidor.	49
Figura 2-3 Modelo de la arquitectura del lado del cliente.	49
Figura 3-1 Diagrama de despliegue.	50
Figura 3-2 Ejemplo de estándares de codificación.	53
Figura 3-3 Pruebas unitarias al sistema.	55
Figura 3-4 Resultado de las pruebas aplicadas.	57

AVAL DEL CLIENTE

OPINION DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, Titulado "**Aplicación Web para el Proceso de Gestión de Contratación y Facturación de Servicios Informáticos de la UEB-Datos de la Empresa de Servicios Especializados GET**" fue realizado en nuestra entidad UEB GET - Datos, perteneciente a la Empresa de Servicios Especializados GET. Se considera que el trabajo realizado satisface y está en "**Total**" correspondencia con los objetivos trazados.

Los resultados de este Trabajo de Diploma le aportan a la entidad los beneficios siguientes:

- Aumenta la organización y el funcionamiento del área comercial en la UEB GET Datos, incrementando la calidad y rapidez en la entrega de los documentos relacionados con el proceso de contratación y facturación.
- Prevé la introducción de posibles errores humanos en la confección de los modelos, planillas e informes oficiales asociados a la actividad comercial.
- Disminución del tiempo de procesamiento de las informaciones y disminución de la carga de trabajo de comerciales de la UEB.
- Se elimina la información redundante y permitiendo, que esta, se almacene en un único lugar con calidad, seguridad y respaldo en una base de datos centralizada y normalizada con diferentes niveles de acceso.

Como resultado de la implantación de este trabajo se reporta un efecto económico considerable, garantizando que el costo hubiera sido mucho mayor si se hubiera optado por la contratación de cualquier software externo.

Y para que así conste, se firme la presente a los **20** día del mes **Noviembre** del año **2023**.

Sandor Michel Calzada Llopiz

Director de la UEB

Director de La UEB GET Datos

Cargo


Firma

INTRODUCCIÓN

En los inicios de este siglo todas las ciencias se han visto influenciadas por el impacto de las nuevas Tecnologías de la Información y las Comunicaciones (TIC) donde la informática juega un papel fundamental en este mundo nuevo, que ha impactado la sociedad. Grandes ramas como la economía y la sociedad se encuentran en pleno desarrollo e implementación de estas tecnologías (Mammadli & Klivak, 2020).

Una de las necesidades a nivel mundial para las empresas es la automatización de sus procesos frente a los nuevos retos que imponen los mercados globalizados, con esto se busca la generación de servicios más ágiles y oportunos a sus clientes; de esta manera se plantea la posibilidad de que las compañías puedan utilizar diferentes mecanismos tecnológicos en procesos de negocio críticos por medio de sistemas de información que permitan garantizar la autenticidad e integridad de los datos y a su vez la entrega, expedición, aceptación y exhibición de la información de manera eficiente, dando cumplimiento a la ley y de esta manera lograr garantizar seguridad jurídica y técnica de los procesos en medios electrónicos (Canizales Muñoz, 2020).

Los avances tecnológicos demandan aplicaciones cada vez más rápidas y robustas que permitan satisfacer las necesidades crecientes de los usuarios. Aprovechando todas las ventajas que brinda la red, se comenzaron a desarrollar aplicaciones de servicio y de resolución de problemas entre las que se encuentran las aplicaciones web. Resultando muy ventajosa su utilización para contribuir a agilizar la gestión de la información. Entre las principales funcionalidades de los sistemas de gestión se encuentra la de procesar la información que se maneja en los distintos procesos empresariales.

La gestión de la contratación con los clientes y la facturación de los servicios contratados son aspectos fundamentales en cualquier empresa, ya que tienen un impacto directo en la generación de ingresos, la optimización de los procesos internos y en la relación con los clientes.

Los servicios informáticos son un conjunto de soluciones tecnológicas y de asistencia especializada que abarcan el diseño, la implementación y el mantenimiento de sistemas informáticos para empresas y organizaciones. Estos servicios se encargan de todo lo referente a la estructura informática de las empresas (Figueroa Morán, 2017).

La Empresa de Servicios Informáticos Especializados, también conocida como Grupo de Electrónica para el Turismo (GET), es una entidad cubana con alcance nacional perteneciente al Ministerio de Turismo, con 28 años ofreciendo servicios de infocomunicaciones y automática a la industria turística en Cuba. Entre los servicios que ofrece GET se encuentran:

- **Software:** Los servicios de software están orientados al desarrollo, comercialización y asistencia técnica de los productos de software. La comercialización se realiza a través de la venta de licencia de uso de software, con o sin valor comercial, aplicándose la modalidad en dependencia de las características del cliente y el producto a adquirir.
- **Desarrollo de software:** Desarrollo de sistemas a la medida o nuevas funcionalidades a los ya existentes, según los requisitos del cliente.
- **Venta de licencia de uso de software:** Venta de licencia con o sin valor comercial, con la que conjuntamente se entrega una copia del ejecutable de cada módulo y la documentación (Manual de usuario y Explotación). Se aplican diferentes modalidades en dependencia de las características del cliente y el producto a adquirir
- **Asistencia técnica:** GET garantiza un servicio ágil y personalizado de soporte técnico presencial y/o remoto a través de las redes de transmisión de datos, teléfono, mensajería, chat o web.
- **Servicios Web:** Facilita la gestión comercial en internet. Ofrece al cliente un servicio completo que incluye diseño, desarrollo, actualización, mantenimiento y posicionamiento en Internet de sitios web hospedado en GET, todo bajo las tendencias actuales de diseño y programación.

También GET ofrece servicios de **hardware, conectividad, automática, datos, seguridad informática y consultoría.**

Entre los servicios informáticos que se encarga de brindar la UEB-Datos se encuentran

- Prestar servicios de transmisión de datos y de Internet.
- Prestar servicios de Comercio Electrónico y efectuar el procesamiento externo y hospedaje de datos para sitios y portales.
- Brindar servicios de asesoría, consultoría, evaluación y dictamen en seguridad informática.

En el área comercial de la UEB-Datos de la Empresa de Servicios Informáticos Especializados GET, el proceso de levantamiento de información durante la contratación de sus clientes y proveedores no se realiza de forma eficiente debido a que la información se maneja de forma manual, con la ayuda del software de oficina Excel, por lo que se genera un gran cúmulo de documentación que debe ser manejada en formato duro, aumentando el riesgo de deterioro, pérdida de la información, dificultades para dar seguimiento al proceso. Además, también da lugar a cometer errores humanos al introducir los datos, de igual manera la variedad de formatos y estructura de la documentación que se genera dificulta la estandarización e intercambio de la información.

A partir de la situación antes descrita se plantea como **problema de la investigación**: ¿Cómo agilizar el proceso de contratación y facturación de servicios informáticos en el área comercial de la UEB - DATOS de la Empresa de Servicios Especializados GET?

Teniendo como **objeto de estudio**: La gestión de contratación y facturación de servicios informáticos. Definiendo, así como **campo de acción**: Sistemas de aplicación web para la gestión de contratación y facturación de servicios informáticos en el área comercial de la UEB - DATOS de la Empresa de Servicios Especializados GET.

Se establece como **objetivo general**: Desarrollar una aplicación web para agilizar el proceso de contratación y facturación de servicios informáticos en el área comercial de la UEB - DATOS de la Empresa de Servicios Especializados GET.

Para dar cumplimiento al objetivo de la investigación, se trazan las siguientes **tareas de la investigación**:

- Definir el marco teórico que permita sustentar una posible propuesta de solución.
- Realizar las actividades y artefactos propuestos por la metodología de desarrollo de software elegida para el desarrollo de la aplicación web de gestión de contratación y facturación de servicios informáticos de la UEB-Datos de la Empresa de Servicios Especializados GET.
- Implementar la aplicación web para la gestión de contratación y facturación de servicios informáticos de la UEB-Datos de la Empresa de Servicios Especializados GET.
- Validar la aplicación web a partir de pruebas de software de tipo unitarias, de sistema y de aceptación.

Para el desarrollo de la aplicación fueron utilizados los siguientes métodos de investigación:

El método **histórico-lógico** se utilizó en el análisis de aplicaciones similares a la propuesta de solución ya existentes.

El método **inductivo-deductivo** fue empleado en la extracción de las ideas fundamentales para la construcción y fundamentación teórica del trabajo de diploma. Además de razonar sobre los temas investigados, con el fin de obtener conocimientos generales y particulares sobre la gestión de contratación y facturación de servicios informáticos.

El método **analítico-sintético** permitió investigar los diferentes elementos asociados a la gestión de contratos y servicios informáticos y realizar una síntesis de la información resultante.

El método de **observación simple** se utilizó en la obtención de información mediante la percepción directa del objeto de la investigación. Permitted la observación del proceso de negocio en el área comercial de la UEB-Datos, con el objetivo de conocer su funcionamiento, organización, estructura de

la información que contiene y las personas que participan y así identificar posibles funcionalidades de la aplicación en el estudio del estado del arte de aplicaciones similares a la propuesta de solución.

El documento está estructurado en tres capítulos, que se detallan a continuación:

Capítulo I: Fundamentación Teórica. Este capítulo contiene los fundamentos teóricos que sustentan al sistema de gestión de contratación y facturación de servicios informáticos en el área comercial de la UEB - DATOS de la Empresa de Servicios Especializados GET. En el mismo se hace un estudio de sistemas homólogos, se describen los principales conceptos para comprender el dominio del problema y se proponen la metodología, tecnologías y herramientas que se empleará en la implementación de la propuesta de solución.

Capítulo II: Diseño de la propuesta de solución al problema de investigación. En este capítulo se explica la propuesta de solución para resolver el problema planteado. Se especifican los requisitos funcionales y no funcionales y los elementos fundamentales del diseño y de la arquitectura que se deben tener en cuenta para el desarrollo del sistema. Por otra parte se establecen los patrones de diseño que se tuvieron en cuenta para el desarrollo del sistema.

Capítulo III: Validación e implementación de la propuesta de solución. En este capítulo se describen las pautas de codificación utilizadas, además la aplicación se someterá a un proceso de pruebas donde se realizarán pruebas unitarias y pruebas de aceptación con el objetivo de verificar el cumplimiento de los requerimientos establecidos.

CAPÍTULO I: Fundamentación Teórica.

Este capítulo se centra en proporcionar una base teórica sólida que respalde y justifique la investigación realizada. Estará dedicado a la búsqueda y análisis de la información relacionada con el dominio del objetivo general. Se comienza sintetizando algunas conceptualizaciones fundamentales relacionadas a los sistemas de gestión comerciales. Se analizan algunas soluciones existentes referentes al mismo entorno. Para finalizar, se realiza la selección y descripción de las principales tecnologías, metodologías y herramientas que serán utilizadas en el desarrollo de la propuesta de solución.

1.1 Definición de los principales conceptos relacionados con el objeto de estudio y campo de acción.

Con el objetivo de lograr un entendimiento del contenido a tratar en la investigación, se requiere el conocimiento de los **conceptos** enunciados a continuación:

Como se mencionaba en la Introducción, los **servicios informáticos** son un conjunto de actividades y soluciones tecnológicas ofrecidas por profesionales o empresas especializadas en tecnología de la información (TI) que abarcan el diseño, la implementación y el mantenimiento de sistemas informáticos, redes y recursos tecnológicos para empresas y organizaciones. Estos servicios se encargan de todo lo referente a la estructura informática de las empresas (Figueroa Morán, 2017).

Los servicios informáticos abarcan una amplia gama de actividades, que pueden incluir:

1. **Desarrollo de software:** Creación, personalización y mantenimiento de aplicaciones informáticas para satisfacer necesidades específicas de una organización.
2. **Soporte técnico:** El servicio de atención al cliente y asistencia técnica para resolver problemas, responder preguntas y brindar orientación sobre el uso de hardware, software o redes.
3. **Servicios de redes:** Configuración, administración y seguridad de redes informáticas, incluyendo conexiones, enrutamiento, cortafuegos, servidores y protocolos de comunicación.
4. **Administración de bases de datos:** Diseño, implementación y gestión de bases de datos, incluyendo la organización de datos, consultas, copias de seguridad, recuperación y seguridad.
5. **Virtualización y almacenamiento:** Implementación y administración de tecnologías de virtualización de servidores y almacenamiento de datos para optimizar el rendimiento y la eficiencia de los sistemas.
6. **Seguridad informática:** Protección de sistemas, datos y redes contra amenazas y ataques cibernéticos, implementación de controles de seguridad, monitoreo y respuesta a incidentes.

7. Consultoría en TI: Asesoramiento y orientación sobre estrategias y soluciones tecnológicas para mejorar la eficiencia, productividad y seguridad de una organización.

Los servicios informáticos son fundamentales en la era digital, ya que permiten a las organizaciones aprovechar al máximo la tecnología para lograr sus objetivos y mantenerse competitivas en el mercado. Estos servicios pueden ser brindados por consultoras de TI, proveedores de servicios especializados o departamentos de TI internos en las organizaciones (Figueroa Morán, 2017).

Los **procesos de gestión** son actividades organizadas y planificadas que se llevan a cabo para lograr un objetivo específico dentro de una empresa u organización. Estos procesos incluyen la planificación, la organización, la coordinación, la supervisión y el control de las actividades y recursos necesarios para alcanzar los objetivos establecidos (Orellana, 2019). Abarcan áreas como la gestión de proyectos, la gestión de recursos humanos, la gestión financiera, la gestión de operaciones, la gestión de calidad, entre otros. Estos procesos son fundamentales para el buen funcionamiento y la eficiencia de una organización, ya que permiten que se lleven a cabo las tareas necesarias de manera ordenada y eficaz (Orellana, 2019).

La **gestión de contratos** es el proceso de administrar y monitorear de manera centralizada cada una de las etapas que constituyen su ciclo de vida. Este proceso incluye la solicitud, redacción, revisión y firma del contrato, hasta su término o renovación (MacNeil, 2022).

La gestión de contratación es un proceso esencial para cualquier empresa que busca controlar, optimizar y tener el máximo control sobre sus acuerdos comerciales (MacNeil, 2022).

Las principales etapas de la gestión de contratación son las siguientes (MacNeil, 2022):

- **Planificación:** En esta etapa, se definen los objetivos del contrato, las partes involucradas y los recursos necesarios.
- **Preparación:** En esta etapa, se recopila la información necesaria para redactar el contrato, como los requisitos del negocio, las condiciones del mercado y la legislación aplicable.
- **Negociación:** En esta etapa, las partes involucradas discuten y negocian los términos del contrato.
- **Elaboración:** En esta etapa, se redacta el contrato final, que incluye todos los términos acordados por las partes.
- **Aprobación:** En esta etapa, el contrato se aprueba por las partes involucradas.
- **Firma:** En esta etapa, el contrato se firma por las partes involucradas.
- **Implementación:** En esta etapa, se pone en marcha el contrato.
- **Seguimiento:** En esta etapa, se supervisa el cumplimiento del contrato.

Una **Proforma de contrato** es un documento que establece los términos y condiciones de un contrato, pero que no es legalmente vinculante. Se utiliza para proporcionar a las partes involucradas una visión general de los términos del contrato antes de firmarlo (Villaruel, 2016), permitiendo regularizar en todas las UEB GET los términos y condiciones fundamentales para la ejecución de los servicios en GET. Las proformas de contrato pueden ser un recurso valioso para las empresas que desean formalizar un acuerdo comercial, pero que no están listas para firmar un contrato legalmente vinculante. Las proformas de contrato pueden ayudar a las empresas a evitar malentendidos y a asegurarse de que todos los términos del contrato están claros y acordados por ambas partes (Villaruel, 2016).

Un **contrato** es un documento que formaliza legalmente las obligaciones y derechos asumidos por las partes signatarias en función del cumplimiento de un objeto. Las partes involucradas pueden ser físicas o jurídicas, y cada parte puede estar constituida por más de una persona. El contrato puede ser de carácter oral o escrito, y puede tener diferentes tipos según la fecha, duración, número de partes, beneficio y composición del contrato (Gil, 2016). Son importantes porque ayudan a proteger los derechos de las partes involucradas. Cuando las partes firman un contrato, están acordando cumplir con los términos del contrato. Si una de las partes no cumple con sus obligaciones, la otra parte puede tomar medidas legales para obligar a la parte incumplida a cumplir (Gil, 2016).

Un **suplemento** es un documento que se une al contrato después de otorgado, para dejar constancia de cualquier modificación o concreción de su contenido, prorrogar su vigencia o declarar su resolución (Cuba. Ministerio de Justicia, 2013). Los suplementos se utilizan cuando las partes desean realizar cambios o agregar cláusulas específicas que no estaban contempladas en el documento original (Cuba. Ministerio de Justicia, 2013). Al utilizar suplementos, se evita tener que redactar un nuevo contrato completo, lo que puede agilizar el proceso de negociación y modificación de los términos, además, es recomendable que los suplementos se adjunten al contrato original para mantener un registro completo de las modificaciones realizadas. Pueden abordar una amplia variedad de aspectos en un contrato, como modificar los términos de pago, establecer nuevas fechas de entrega, agregar o eliminar servicios o productos, realizar ajustes en los precios o tarifas, entre otros.

El **reporte de producción** es un documento que permite conocer los resultados y datos de los procesos y datos de producción. Es una herramienta muy efectiva para monitorear en tiempo real la producción, costos, rendimiento y calidad. El reporte de producción permite conocer resultados con certeza de los procesos y datos de producción (Milformatos, 2022). Tiene como objetivo presentar de manera detallada los servicios a facturar, en correspondencia con su ejecución, minimizando las diferencias entre lo contratado y lo facturado; así como determinar el pronóstico de cumplimiento del plan de ingresos del mes que se tributa de manera semanal a la reunión de puntualización.

La **factura** es un documento comercial que se emite con la descripción, cantidad e importes de los servicios brindados, en correspondencia con el objeto y términos establecidos en el contrato. La facturación se realiza de acuerdo al reporte de producción (Frederick, 2023).

A pesar de que no existe un modelo estándar, todas ellas deben incluir una serie de contenidos mínimos:

- **Lugar y fecha de emisión:** En todas las facturas debe indicarse el lugar y la fecha en la que se emite. Habitualmente, se indica en el encabezamiento.
- **Numeración de factura:** Junto al lugar y la fecha, se ha de indicar la numeración de la factura. Esta numeración es cronológica en función de la facturación del emisor.
- **Identificación de comprador y vendedor:** debe identificarse al comprador y el vendedor.
- **Descripción de la operación:** En el cuerpo de la factura, se realiza una breve descripción de la operación u operaciones que forman parte de la factura.
- **Importe:** El importe total del servicio, de acuerdo a la cantidad a facturar
- **Importe Total:** El importe total que se va a facturar en el mes

1.2 Sistemas Homólogos.

Los programas de gestión de contratación y facturación de servicios informáticos son un aliado muy valioso para conocer de primera mano lo que está sucediendo en nuestra empresa a nivel financiero.

En el estudio realizado a algunos de los sistemas existentes de gestión comercial utilizados de forma global y dentro del país. Se analizaron varios factores fundamentales, como la gestión de la información referente a la contratación, facturación y oferta comercial. A continuación, se describen los sistemas estudiados.

Odoo Este ERP¹ (Enterprise Resource Planning) open source destaca por ser muy modular y flexible, fácilmente ajustable a cada empresa en particular. El programa de facturación de Odoo se puede contratar junto con el de contabilidad, o bien añadirlo a módulos de otras áreas para integrar toda la información empresarial. Cuenta con una demostración gratuita que permite ver su potencial en la gestión fácil de contratos, la creación de facturas recurrentes y la automatización de transacciones habituales como cobros y pagos (Muñoz, 2023).

¹ Los ERP son los sistemas de gestión más conocidos. Lo utilizan las empresas para administrar de forma integral las diferentes actividades diarias como la contabilidad, el abastecimiento, la administración de proyectos, el cumplimiento y la gestión de riesgos y las operaciones de la cadena de suministros.

El **ERP Zoom Localización Cubana de Odoo** está desarrollado sobre la base del software libre Odoo, tecnología open source con más de 3 millones de usuarios y una amplia comunidad internacional de respaldo. El sistema, adaptado a las normas y regulaciones nacionales fundamentalmente en materia contable financiera, integra más de 30 módulos o aplicaciones para la gestión de diferentes áreas de un negocio.

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

Tabla 1-1 Pros y Contras de Odoo.
Fuente: Elaboración Propia.

Pros	Contras
Precios muy asequibles y modulables	Dificultad de controlar lo que gastas con los extras
Posibilidad de crear módulos extra	Interfaz de usuario mejorable

Holded es un software inteligente de gestión de empresas, especialmente diseñado para autónomos y pymes. Cuenta con un diseño y aspecto enfocado en la usabilidad y utilidad y entre sus funciones se encuentran: facturación, contabilidad, proyectos, CRM y gestión de inventario. Es posible generar informes automáticos con todos los datos de tus facturas. Se podrá consultar las pérdidas y ganancias en tiempo real, haciendo comparaciones con los objetivos planteados. De esta manera, se hace posible hacer proyecciones a futuro de cómo podría evolucionar la empresa.

Holded permite conectarse también con otras herramientas que amplían las posibilidades de los usuarios, como PayPal, Contabilidad A3 y Stripe.

En general, permite obtener una visión general de todas las gestiones de una pyme, incluyendo la facturación, de forma sencilla y rápida.

Holden tiene entre sus principales funcionalidades:

- Profesionalizar los servicios con plantillas que se adapten a las necesidades.
- Cobrar de manera rápida gracias a su compatibilidad con cientos de bancos.

- Visualizar las finanzas al tener toda la información en un solo lugar.
- Conciliar facturas y transacciones bancarias en tiempo récord.
- Ver informes detallados sobre gastos y movimientos.

(Muñoz, 2023)

Tabla 1-2 Pros y Contras de Holden.
Fuente: Elaboración Propia.

Pros	Contras
Muy fácil de aprender a usar y cuenta con una enorme guía de ayuda en la Academy	Su cálculo de impuestos solo está enfocado a España
Unifica facturación y toda la gestión de la empresa en un solo lugar	No hay versión gratuita enfocada a pymes, solo a autónomos

Distra es un sistema de gestión empresarial, que permite integrar y optimizar los procesos empresariales, manejar con eficacia la información y disminuir los costos de las operaciones. Con su uso se puede manejar, de forma centralizada, la información relevante de las estructuras organizativas e integrar todos los procesos de trabajo. Este es un sistema cubano desarrollado en la XETID, empresa de referencia destinada al desarrollo de tecnologías de la información para la defensa.

Está compuesto por diversos subsistemas y módulos, que cubren las principales áreas de gestión de cualquiera entidad:

- DM Datos maestros
- GO Gestión organizacional
- PF Planificación
- CH Capital humano (implementa normas cubanas 3000, 3001 y 3002)
- CF Contabilidad y finanzas
- CS Cadena de suministro
- GC Gestión comercial
- GI Gestión de inventario
- FA Gestión de Facturación
- DI Distribución
- LO Logística
- AF Activos fijos tangibles

Es una oferta muy útil para las medianas y grandes empresas, que poseen varias entidades subordinadas y para aplicar el teletrabajo, en cualquiera de las áreas de gestión de la entidad (Xetid, 2021).

Quipu es un software de gestión empresarial en la nube mediante el cual se puede gestionar la facturación, impuestos, tesorería y bancos. Es un programa con más de 9 años de experiencia en el sector y es reconocido como el programa que las pymes más recomiendan para la gestión empresarial. Ofrece seguridad en sus servidores para guardar información sensible como lo es la de facturación y de forma sencilla optimiza tareas y procesos administrativos diarios.

Este software de gestión administrativa tiene la gran ventaja de trabajar con “gestorías certificadas” que se puede añadir al plan. Actualmente cuentan con más de 300 gestorías en España que se han certificado en el uso de la plataforma y se conectan al programa para ayudarte en la parte más difícil de la contabilidad; balances, cierres anuales, pérdidas y ganancias, presentación de impuestos y, obviamente, ofrecen asesoramiento fiscal-contable para el negocio (Quipu, 2023).

1.2.1 Análisis específico de los sistemas homólogos.

Para un mejor análisis de los sistemas estudiados se desarrolla la siguiente tabla:

*Tabla 1-3 Análisis específico de los sistemas homólogos.
Fuente: Elaboración Propia.*

Programa	Almacenamiento	Contrato	Facturación	Oferta Comercial	Licencia
Oddo	En la nube	Sí	Sí	Sí	De pago
Holded	En la nube	No	Sí	No	De pago
Distra	En la nube	Sí	Sí	Sí	De pago
Quipu	En la nube	No	Sí	No	De pago

1.2.2 Valoración crítica de los sistemas estudiados

Los sistemas estudiados de gestión de contratación y facturación, son software de pago, que no generan los artefactos necesarios para una correcta gestión de los procesos de contratación y facturación en el área comercial de la UEB-Datos de la Empresa de Servicios Especializados GET, así como tampoco se rigen por las mismas normas y esquemas requeridas, están asociadas a instituciones u organizaciones privadas o extranjeras por lo que no cumplen con la política de soberanía del país y

no son intuitivas ni fáciles de utilizar incluso para los usuarios más expertos necesitando preparación técnica previa. De ahí que no sea posible dar soporte, ni personalizarlas a las características propias del centro. Por lo que se decide llevar a cabo la realización de este proyecto.

1.3 Ambiente de desarrollo

Un ambiente de desarrollo es un conjunto de procedimientos y herramientas que se utilizan para desarrollar un código fuente o programa. Este término se utiliza a veces como sinónimo de entorno de desarrollo integrado (IDE), que es la herramienta de desarrollo de software utilizado para escribir, generar, probar y depurar un programa. La administración de ambientes de desarrollo de software es importante para garantizar que los desarrolladores de software, tanto internos como externos a la organización, compartan los mismos ambientes de desarrollo y sigan procedimientos adecuados para administrarlos. Algunas buenas prácticas para la administración de ambientes de desarrollo incluyen la definición de las características adecuadas de infraestructura, base de datos, organización, planeación y procedimientos de cambios, homologación, altas y bajas (PMOInformatica, 2012).

1.3.1 Metodología de desarrollo de software

Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo (Sommerville, 2019). La selección y uso de la metodología adecuada es uno de los puntos vitales para el éxito de un proyecto. Esta dicta qué hacer y cómo hacerlo, es el cimiento que sostendrá todo el proceso de desarrollo posterior, por lo que resulta crítico tener buen juicio al seleccionar la que mejor se adapte a las características del proyecto en cuestión. Las metodologías de desarrollo de software suelen clasificarse según sus características fundamentales y el enfoque utilizado. Hasta el momento han sido identificados dos grandes grupos: metodologías tradicionales o prescriptivas y metodologías ágiles o ligeras (Beck, 2018).

La metodología escogida para el proceso de desarrollo es **SCRUM** debido a que esta herramienta tributa al mecanismo de desarrollo de la empresa GET, la cual tiene un enfoque ligero se centra en los miembros del equipo y su interacción, en la entrega rápida de versiones de software funcional, en la colaboración constante del cliente y la facilidad para manejar los cambios, dándole menor importancia a las herramientas, documentación, formalidad y planificación exhaustiva del proceso.

Esta metodología se dedica al desarrollo del producto con el objetivo de que progrese de manera constante. En vez de basarse en una planificación completa, su desarrollo se revisa periódicamente. Se caracteriza por apoyar la colaboración y la autoorganización, autogestionar las actividades del equipo Scrum y entregar versiones funcionales del producto final continuamente a lo largo de ciclos incrementales (sprints) que transcurren semanalmente o, como máximo, de mes en mes. Scrum se divide en 5 fases: Inicio, Planificación y estimación, Implementación, Revisión y retrospectiva y Lanzamiento (Beck, 2018).

Miembros de un equipo de scrum

Un equipo de scrum es un equipo pequeño y ágil que se dedica a ofrecer incrementos de productos de forma comprometida. El tamaño de un equipo de scrum suele ser pequeño, de unas 10 personas, pero es lo suficientemente grande como para completar una cantidad sustancial de trabajo en un sprint. El equipo de scrum debe componerse de tres cargos específicos: el propietario del producto, el experto en scrum y el equipo de desarrollo. Y, puesto que los equipos de scrum son interdisciplinarios, el equipo de desarrollo está formado por evaluadores, diseñadores, especialistas en experiencia de usuario e ingenieros de operaciones, además de desarrolladores (Drumond, 2023).

Scrum define tres roles, los cuales son (Beck, 2018):

- El dueño del producto o Product Owner, es el representante de los accionistas y clientes que usan el software.
- El Scrum Master, es la persona que lidera el equipo asegurándose que el equipo cumpla las reglas y procesos de la metodología.
- El equipo de desarrollo o team, es el grupo de profesionales encargados de convertir la lista de requerimientos o también llamado Product Backlog en funcionalidades del software.

Artefactos de scrum

Los artefactos de scrum ofrecen información importante que el equipo de scrum utiliza para definir el producto y el trabajo que hay que hacer para crearlo. Existen tres artefactos en scrum: un backlog del producto, un backlog de sprint y un incremento en la definición de "finalizado". Son las tres constantes sobre las que un equipo de scrum debe reflexionar durante los Sprints y a lo largo del tiempo (Drumond, 2023).

- **Backlog del producto:** es la lista principal del trabajo que debe realizar el propietario o el responsable del producto. Se trata de una lista dinámica de funciones, requisitos, mejoras y

correcciones que actúa como la base del backlog de sprint. Básicamente, se trata de la lista de "tareas que hay que hacer" del equipo. El propietario del producto está constantemente revisando, cambiando las prioridades y realizando el mantenimiento del backlog del producto, ya que, a medida que sabemos más o que cambia el mercado, es posible que haya elementos que ya no sean relevantes o que los problemas se solucionen de otras maneras.

- **Backlog de sprint:** se trata de la lista de elementos, historias de usuario o correcciones de errores, seleccionadas por el equipo de desarrollo, para su implementación en el ciclo actual de sprint. Antes de cada sprint, en la reunión de planificación de sprint, el equipo elige los elementos en los que trabajará para el sprint del backlog del producto. El backlog de sprint puede ser flexible y puede evolucionar durante un sprint. No obstante, no se puede poner en peligro el objetivo fundamental del sprint, lo que el equipo quiere lograr con el sprint actual.
- **Incremento (u objetivo del sprint):** es el producto final utilizable de un sprint. No se suele escuchar la palabra "incremento" en ningún sitio, ya que a menudo se la conoce como la definición del equipo de "Finalizado", un hito, el objetivo del sprint o incluso una versión completa o un epic lanzado. Solo depende de la definición de "Finalizado" de los equipos y de cómo cada uno define sus objetivos del sprint.

1.3.2 Caracterización de las tecnologías y las herramientas

Lenguajes de programación

Un lenguaje de programación es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones (Aprendiendo Arduino, 2017).

Marco de trabajo

Entorno de trabajo o marco de trabajo del inglés *framework* es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar (Gandarillas, 2017). Es la forma de agrupar varios conceptos comunes, prácticas y criterios para enfrentar una problemática, convirtiéndolo en una referencia para resolver posibles problemas asociadas a esa problemática, o aquellos similares. Específicamente en el desarrollo de software se agrega un componente tecnológico que permite que sea la base de la construcción del software. Un marco proporciona una gama flexible de componentes

de software que ayudan a los desarrolladores a acelerar el desarrollo del software hasta su despliegue en la producción. Consiste en recursos creados para un lenguaje de programación, como las API, las bibliotecas de códigos, los depuradores y los compiladores (Lago, 2022).

Los marcos de trabajo permiten (Gandarillas, 2017):

- Facilitar el desarrollo de software.
- Evitar los detalles de bajo nivel, permitiendo concentrar más esfuerzo y tiempo en identificar los requerimientos de software.

Tecnologías del lado del cliente

Las tecnologías del lado del cliente son la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que se ve en la pantalla cuando se accede a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas (RWD), los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales entre otros elementos que permiten navegar dentro de una página web. Este conjunto crea la experiencia del usuario (Stefaniak, 2019).

Para la selección del marco de trabajo o librerías, en el **lado del cliente**, se tuvieron en cuenta los siguientes:

- **Vue.js** es un marco de trabajo progresivo lanzado en 2014 para construir interfaces de usuario. A diferencia de otros marcos de trabajo monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización y es fácil de utilizar e integrar con otras librerías o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar sofisticadas *Single-Page Applications*² cuando se utiliza en combinación con herramientas modernas y librerías de apoyo.
- **React.js** es una biblioteca de JavaScript para construir interfaces de usuario lanzada en 2013 por Facebook. Es altamente dinámico y ofrece un gran soporte en la creación de interfaces de usuario interactivas. No es específico del navegador y ligero. Es gratuito y elimina los problemas de rendimiento lento del DOM real ya que usa un DOM virtual. Esto permitió una mejora significativa en el rendimiento de los *marcos de trabajo* y librerías de JavaScript e hizo que React fuera completamente popular (React, 2023).
- **Angular.js** es una plataforma y un marco de trabajo para crear aplicaciones de una sola página en el lado del cliente usando HTML y TypeScript. Angular está escrito en TypeScript.

² Single-Page Application (SPA) o Aplicación de Página Única, es un tipo de aplicación web que ejecuta todo su contenido en una sola página.

Implementa la funcionalidad básica y opcional como un conjunto de bibliotecas TypeScript que importas en tus aplicaciones (Angular, 2023).

Tabla 1-4 Comparación entre Angular, React y Vue.

Fuente: Elaboración Propia.

Atributos	Angular	React	Vue
Tipo	Marco de trabajo de JavaScript	Biblioteca de Código Abierto de JS	Marco de trabajo de JavaScript Progresivo
Descargas semanales Npm (2023)	9,102,900	22,607,472	4,212,300
Tamaño	696 kB	316 kB	2.02 MB
Fácil de aprender	Difícil (Aprender TypeScript)	Moderado	Fácil
Velocidad al escribir código	Lento	Normal	Rápido
Documentación	Sí	Sí	Sí
Rendimiento	Bueno	Bueno	Bueno
Tiempo de Inicio	Más larga debido a su gran base de código	Rápido	Rápido
Aplicaciones web completas	Se puede utilizar de forma independiente	Necesita integrarse con muchas otras herramientas	Requiere herramientas de terceros
Vinculación de datos	Bi-Direccional	Uni-Direccional	Bi-Direccional
Renderizado	Lado del cliente	Lado del servidor	Lado del servidor
Modelo	MVC	Virtual	Virtual
Reutilización del código	Sí	No, solo CSS	Yes, CSS y HTML
Cuando usar	En producción especialmente aplicaciones	En producción, aplicaciones de interfaz de usuario	Startup ³ , en producción

³ Un startup es una empresa de nueva creación que, gracias a su modelo de negocio escalable y al uso de las nuevas tecnologías, tiene grandes posibilidades de crecimiento (¿Qué Es Una “Startup”?, 2022).

	empresariales con material ui	personalizadas	
--	-------------------------------	----------------	--

Estos marcos de trabajo guardan cierta similitud en cuanto a rendimiento y flexibilidad, son adecuados para aplicaciones livianas y gozan de mucha popularidad en la comunidad de programadores. Se eligió para la aplicación el marco de trabajo Vue por su simplicidad y eficiencia facilita la creación de aplicaciones e interfaces de usuario en forma rápida, práctica y relativamente simple.

Para complementar el trabajo con Vue se usó “**Vuetify**” que es un marco de interfaz de usuario completo construido sobre Vue.js. El objetivo de Vuetify es proporcionar a los desarrolladores las herramientas necesarias para crear experiencias de usuario ricas y atractivas. A diferencia de otros marcos, Vuetify está diseñado desde cero para ser fácil de aprender y gratificante de dominar con cientos de componentes cuidadosamente elaborados a partir de la especificación Material Design. Vuetify adopta un primer enfoque móvil para el diseño, lo que significa que su aplicación funciona de inmediato, ya sea en un teléfono, tableta o computadora de escritorio (Vuetify 2, 2023).

Lenguajes usados del lado del cliente

HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto. Se trata de un formato abierto que surgió a partir de las etiquetas SGML (Standard Generalized Markup Language). Concepto traducido generalmente como «Estándar de Lenguaje de Marcado Generalizado» y que se entiende como un sistema que permite ordenar y etiquetar diversos documentos dentro de una lista. Este lenguaje es el que se utiliza para especificar los nombres de las etiquetas que se utilizarán al ordenar, no existen reglas para dicha organización, por eso se dice que es un sistema de formato abierto. El HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos (como fotografías, animaciones, entre otros) (Meloni & Kyrnin, 2019).

CSS sus siglas corresponden a la expresión inglesa Cascading StyleSheets, que puede traducirse como “Hojas de estilo en cascada”. El concepto se utiliza en el ámbito de la informática para referirse a un lenguaje empleado en el diseño gráfico. El lenguaje CSS permite presentar, de manera estructurada, un documento que fue escrito en un lenguaje de marcado. Se usa especialmente en el diseño visual de un sitio web cuando las páginas se hallan escritas en XML o HTML. El diseño del CSS

posibilita establecer una separación entre el contenido y la forma de presentación del documento (dada por las fuentes, los colores y las capas empleadas). Así se puede lograr que muchos documentos HTML compartan la apariencia, utilizando una única hoja de estilo para todos (que se especifica en un archivo .css) así se evita tener que repetir el código en la estructura. Gracias a la separación del contenido y la forma de presentación, por otra parte, se puede apelar a distintos estilos de acuerdo al método de renderizado: si se trata de un documento exhibido en una pantalla, impreso, compartido en formato de audio, entre otros (Meloni & Kyrnin, 2019).

JavaScript (o "JS") es un lenguaje de programación que se usa con mayor frecuencia para scripts dinámicos de lado del cliente en páginas web, pero también se usa a menudo en el lado del servidor usando un entorno de ejecución como Node.js. JavaScript no se debe confundir con el lenguaje de programación Java. Aunque "Java" y "JavaScript" son marcas comerciales (o marcas comerciales registradas) de Oracle en EE.UU. y otros países, los dos lenguajes de programación son significativamente diferentes en su sintaxis, semántica y casos de uso. JavaScript se utiliza principalmente en el navegador, lo que permite a los desarrolladores manipular el contenido de la página web a través del DOM, manipular datos con AJAX y IndexedDB, dibujar gráficos con canvas, interactuar con el dispositivo que ejecuta el navegador a través de varias APIs y más. JavaScript es uno de los lenguajes más utilizados en el mundo, debido al reciente crecimiento y mejora en el rendimiento de las APIs disponibles en los navegadores (Meloni & Kyrnin, 2019)

Tecnologías del lado del servidor

En el mundo de la informática, las tecnologías del lado del servidor (*backend*) se refiere a cualquier parte de un sitio web o programa de software que los usuarios no ven. Contrasta con la interfaz, que se refiere a la interfaz de usuario de un programa o sitio web. En terminología de programación, las tecnologías del lado del servidor son la "capa de acceso a datos". La mayoría de los sitios web modernos son dinámicos, lo que significa que el contenido de la página web se genera sobre la marcha. Una página dinámica contiene uno o más scripts que se ejecutan en el servidor web cada vez que se accede a la página. Estos scripts generan el contenido de la página, que se envía al navegador web del usuario. Todo lo que sucede antes de que la página se muestre en un navegador web es parte del backend (TechTerms.com, 2020).

En cuanto a las **tecnologías del lado del servidor** (*backend*), se decidió reutilizar el lenguaje seleccionado en las tecnologías del lado del cliente (JavaScript) usando como entorno de ejecución Node.js

Node.js es un entorno de tiempo de ejecución multiplataforma y de código abierto basado en el motor JavaScript V8 de Chrome para ejecutar código JavaScript fuera de un navegador. Proporciona un entorno de tiempo de ejecución multiplataforma y de E/S sin bloqueo (asíncronico) basado en eventos para crear aplicaciones del lado del servidor altamente escalables mediante JavaScript (Morales, 2019). Node.js es usado usualmente para crear servicios como API, aplicaciones web o aplicaciones móviles. Es utilizado en la producción por grandes empresas como PayPal, Uber, Netflix, Walmart, entre otros.

Algunas de las **características más destacables de Node.js** son (Heddings, 2020):

1. Fácil – Node.js es bastante fácil de empezar. Es una opción para los principiantes en el desarrollo web. Con un montón de tutoriales y una gran comunidad.
2. Escalable – proporciona una gran escalabilidad para las aplicaciones. Node.js, al ser de un solo hilo, es capaz de manejar un gran número de conexiones simultáneas con un alto rendimiento.
3. Velocidad – La ejecución de hilos sin bloqueo hace que Node.js sea aún más rápido y eficiente.
4. Paquetes – existe un amplio conjunto de paquetes de Node.js de código abierto que pueden simplificar el trabajo. Actualmente hay más de un millón de paquetes en el ecosistema NPM.
5. Sólido respaldo – Node.js está escrito en C y C++, lo que le confiere rapidez y añade características como la compatibilidad con redes.
6. Multiplataforma – el soporte multiplataforma le permite crear sitios web SaaS, aplicaciones de escritorio e incluso aplicaciones móviles, todo ello utilizando Node.js.
7. Mantenible – Node.js es una opción fácil para los desarrolladores ya que tanto las tecnologías del lado del cliente como las tecnologías del lado del servidor pueden ser gestionados con JavaScript como un único lenguaje.

API

El término API es una abreviatura de *Application Programming Interfaces*, que en español significa interfaz de programación de aplicaciones. Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a partir de un conjunto de reglas. Una API es una especificación formal que establece cómo un módulo de un software se comunica o interactúa con otro para cumplir una o muchas funciones. Todo dependiendo de las aplicaciones que las vayan a utilizar y de los permisos que les dé el propietario de la API a los desarrolladores de terceros (Núñez, 2021).

Express.js Express.js se utiliza para crear API (interfaz de programación de aplicaciones) y aplicaciones móviles. Se encarga de los detalles vitales de las tecnologías del lado del servidor, como las sesiones, la gestión de errores y el enrutamiento (Green & Green, 2016). Es el marco de trabajo

más popular para las tecnologías del lado del servidor para Node.js y es una parte extensa del ecosistema JavaScript.

El marco de trabajo proporciona un conjunto de herramientas para aplicaciones web, peticiones y respuestas HTTP, enrutamiento y middleware para construir y desplegar aplicaciones a gran escala y preparadas para la empresa.

El elemento principal de la solución será un servicio en forma de API, que permita luego ser consumida por la Interfaz Web.

Sistema Gestor de Base de Datos

PostgreSQL es un servidor de base de datos relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, liberado bajo la licencia BSD (Berkeley Software Distribution). Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola compañía, sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). SQL es el lenguaje que PostgreSQL (y la mayoría del resto de bases de datos relacionales) usa como lenguaje de consultas. Es portable y fácil de aprender. Pero cada sentencia SQL debe ser ejecutada individualmente por el servidor de bases de datos. Esto significa que su aplicación cliente debe enviar cada consulta al servidor de bases de datos, esperar a que se procese, recibir el resultado, realizar alguna computación y luego enviar otras consultas al servidor. Todo esto incurre en una comunicación entre procesos y también puede sobrecargar la red si su cliente se encuentra en una máquina distinta al servidor de bases de datos (Postgres team, 2023).

Las características positivas que posee este gestor son:

1. Posee una gran escalabilidad. Es capaz de ajustarse al número de CPU y a la cantidad de memoria que posee el sistema de forma óptima, haciéndolo capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.
2. Implementa el uso de rollbacks⁴, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz y ofreciendo soluciones en campos en los que MySQL no podría.

⁴La sentencia ROLLBACK se utiliza para restituir los cambios que se han hecho en la base de datos dentro de una unidad de trabajo o punto de salvaguarda. (IBM, 2023)

3. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.

Herramientas de desarrollo

Entorno integrado de desarrollo (IDE)

Un entorno de desarrollo integrado (IDE) es un sistema de software para el diseño de aplicaciones que combina herramientas del desarrollador comunes en una sola interfaz gráfica de usuario (GUI). Generalmente, un IDE cuenta con las siguientes características (Moreno Pérez, 2018):

- Editor de código fuente: editor de texto que ayuda a escribir el código de software con funciones como el resaltado de la sintaxis con indicaciones visuales, el relleno automático específico para el lenguaje y la comprobación de errores a medida que se escribe el código.
- Automatización de las compilaciones locales: herramientas que automatizan las tareas sencillas y repetitivas como parte de la creación de una compilación local del software para que use el desarrollador, como la compilación del código fuente de la computadora en código binario, el empaquetado de ese código y la ejecución de pruebas automatizadas.
- Depurador: programa que sirve para probar otros programas y mostrar la ubicación de un error en el código original de forma gráfica.

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C++, C#, Java, Python, PHP, Go, .NET) (Johnson, 2019)

Algunas de las características de Visual Studio Code son:

- **Multiplataforma**: Visual Studio Code está disponible en Windows, GNU/Linux y macOS.
- **IntelliSense**: Esta característica está relacionada con la edición de código, autocompletado y resaltado de sintaxis, lo que permite ser más ágil a la hora de escribir código.
- **Depuración**: Visual Studio Code cuenta con soporte para depuración de código, lo que facilita la identificación y corrección de errores.

- **Extensiones:** Visual Studio Code dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación.
- **Interfaz de usuario personalizable:** Visual Studio Code ofrece una interfaz de usuario intuitiva y fácil de usar. Es sencilla y cómoda. Su enfoque minimalista permite una experiencia de desarrollo limpia y sin distracciones.

PgAdmin 4 es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación *Slony-I*⁵ y mucho más (Postgres team, 2023).

Git es un sistema de control de versiones distribuido que se utiliza para rastrear cambios en el código fuente durante el desarrollo de software. Permite a los desarrolladores colaborar en un proyecto y mantener un historial completo de los cambios realizados en el código. Fue creado por Linus Torvalds en 2005 para ayudar en el desarrollo del kernel de Linux. Es una herramienta muy popular entre los desarrolladores debido a su velocidad, eficiencia y flexibilidad (Git SCM, 2023).

Conclusiones del capítulo

Luego de ser consultada la bibliografía especializada y obtener un marco teórico y estado del arte para la investigación, se concluye lo siguiente:

- Con el estudio de los conceptos asociados a la gestión de los contratos y las facturas se asentaron las bases para un mejor entendimiento del dominio de la investigación.
- El análisis de aplicaciones que llevan el control de la contratación y la facturación existentes permitió definir un conjunto de características para ser incorporadas a la solución desarrollada.

⁵ Slony-I es un sistema de replicación maestro-esclavo asíncrono para DBMS PostgreSQL, que brinda soporte para cascada y conmutación por error. Asíncrono significa que cuando una transacción de la base de datos se ha enviado al servidor maestro, aún no se garantiza que esté disponible en los esclavos.

- La selección de la metodología SCRUM resulta ideal para guiar el proceso de desarrollo de la aplicación por la organización y la eficiencia con la que se propone cumplir con las tareas para que se lleve a cabo el correcto funcionamiento del sistema.
- El estudio de tecnologías para el desarrollo de software, permitió seleccionar las más adecuadas para la implementación de la solución propuesta.

CAPÍTULO II: Diseño de la propuesta de solución al problema de investigación.

En este capítulo se describe la propuesta de solución y las actividades desarrolladas durante sus procesos de análisis y diseño, tomando como punto inicial los elementos trazados por la metodología SCRUM. Se da a conocer los requerimientos funcionales y no funcionales identificados, las historias de usuarios, se usa de guía los artefactos que dicta la metodología SCRUM, la arquitectura del sistema y los patrones de diseño utilizados.

2.1 Descripción de la solución

En la introducción se planteó las dificultades que se presentaban en el área comercial de la UEB-Datos de la Empresa de Servicios Especializados GET. Tras la presentación de dichas dificultades se propone como solución la creación de una aplicación web, la cual tiene como objetivo brindarles a sus usuarios que puedan gestionar todas las operaciones que realizan a diario y que obtengan un fácil acceso a toda la información referente a los clientes, los servicios y los contratos que estos tienen con la UEB y que puedan generar dichos contratos más las facturas que se les cobra a cada cliente por los servicios brindados.

El sistema cuenta con una serie de módulos donde se puede acceder a las operaciones que se realizan en el área comercial de la UEB-Datos estos son: la gestión de Clientes, Tipos de servicios, Servicios, Solicitud de servicios, Ofertas comerciales, Contratos, Reportes de producción y Facturas.

Esta propuesta de solución se divide en dos partes, un servidor web que ofrece los servicios para enviar y recibir los datos a manejar y una aplicación web que consume los servicios emitidos por el servidor web. Los datos manejados por ambas partes son guardados por el servidor en una base de datos.

En la Figura 2.1 se explica de forma gráfica la propuesta de solución.

Figura 2-1 Propuesta de Solución.
Fuente: Elaboración Propia.



2.3 Requisitos de software

La ingeniería de requisitos es el proceso de desarrollar una especificación de software. Las especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema (Sommerville, 2019).

2.3.1 Requisitos funcionales

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información (Sommerville, 2019).

Los requerimientos funcionales son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema (Sommerville, 2019).

A continuación, en la tabla 2-1 se listan los requerimientos funcionales identificados para el desarrollo de la propuesta de solución.

Tabla 2-1 Requisitos Funcionales.
Fuente: Elaboración Propia.

Nº	Nombre	Descripción	Prioridad para el cliente	Complejidad
RF 1	Autenticar usuario	Permite ingresar al usuario su correo y contraseña en el sistema	Alta	Media
RF2	Registrar usuario	Permitirá añadir un usuario en el sistema con sus datos correspondientes	Alta	Media
RF 3	Editar usuario	Permitirá editar un usuario en el sistema	Media	Media
RF 4	Listar usuarios	Permitirá mostrar en forma de lista los usuarios en el sistema	Media	Media
RF 5	Deshabilitar usuario	Permitirá deshabilitar un usuario en el sistema	Media	Media
RF 6	Añadir cliente	Permitirá insertar un cliente en el sistema con sus datos correspondientes	Alta	Media
RF 7	Editar cliente	Permitirá editar un cliente en el sistema	Media	Media
RF 8	Listar cliente	Permitirá mostrar en forma de lista los clientes en el sistema	Media	Media
RF 9	Añadir Tipo de Servicio	Permitirá insertar un Tipo de Servicio en el sistema con sus datos correspondientes.	Alta	Media
RF 10	Editar Tipo de Servicio	Permitirá editar un Tipo de Servicio en el sistema	Media	Media
RF 11	Listar Tipo de Servicio	Permitirá mostrar en forma de lista los Tipos de Servicios en el sistema	Media	Media
RF 12	Añadir Servicio	Permitirá insertar un servicio en el sistema con sus datos correspondientes	Alta	Media
RF 13	Editar Servicio	Permitirá editar un servicio en	Media	Media

		el sistema		
RF 14	Listar Servicio	Permitirá mostrar en forma de lista los servicios en el sistema	Media	Media
RF 15	Añadir Solicitud de Servicio	Permitirá insertar una Solicitud de Servicio en el sistema con sus datos correspondientes	Alta	Media
RF 16	Editar Solicitud de Servicio	Permitirá editar una Solicitud de Servicio en el sistema	Media	Media
RF 17	Listar Solicitud de Servicio	Permitirá mostrar en forma de lista las Solicitudes de Servicios en el sistema	Media	Media
RF 18	Crear Oferta Comercial	Permite crear una Oferta Comercial a partir de una Solicitud de Servicio añadida al sistema	Alta	Media
RF 19	Editar Oferta Comercial	Permitirá editar una Oferta Comercial en el sistema	Media	Media
RF 20	Listar Oferta Comercial	Permitirá mostrar en forma de lista las Oferta Comerciales en el sistema	Media	Media
RF 21	Imprimir Oferta Comercial	Permitirá imprimir la Oferta Comercial	Alta	Media
RF 22	Crear Contrato	Permite crear un Contrato en el sistema	Alta	Media
RF 23	Listar Contrato	Permitirá mostrar en forma de lista los Contratos en el sistema	Media	Media
RF 24	Imprimir Contrato	Permitirá imprimir los contratos	Alta	Media
RF 25	Crear Suplemento	Permite crear un Suplemento en el sistema	Alta	Media
RF 26	Listar Suplemento	Permitirá mostrar en forma de lista los Suplementos en el sistema	Alta	Media
RF 27	Añadir Reporte de Producción	Permite ingresar al sistema el Reporte de Producción de los servicios brindados en el mes a los clientes	Alta	Media

RF 28	Editar Reporte de Producción	Permitirá editar el Reporte de Producción en el sistema	Media	Media
RF 29	Listar Reporte de Producción	Permitirá mostrar en forma de lista los Reportes de Producción	Media	Media
RF 30	Imprimir Reporte de Producción	Permitirá imprimir las facturas	Alta	Media
RF 31	Imprimir Factura	Permitirá imprimir las facturas	Alta	Media
RF32	Generar el Contrato a imprimir	Permitirá generar el contrato para que sea impreso	Alta	Alta
RF33	Generar la Factura a imprimir	Permitirá generar la factura para que sea impresa, a partir del reporte de producción realizado	Alta	Alta

2.3.2 Requisitos no funcionales

Los requisitos no funcionales explican las limitaciones y restricciones del sistema a diseñar. Estos requisitos no tienen ningún impacto en la funcionalidad de la aplicación. Además, existe una práctica común de subclasificar los requisitos no funcionales en varias categorías. Subclasificar los requisitos no funcionales es una buena práctica ya que ayuda a la hora de crear una lista de verificación de los requisitos que se deben cumplir en el sistema a diseñar (Visure Solutions, n.d.).

- **Interfaz:**

RNF1. El sistema debe incluir el nombre y logo de GET, así como los colores que distinguen a la empresa.

- **Seguridad:**

RNF2. La información que se maneje se mantendrá segura y confidencial, evitando el acceso no autorizado y la divulgación mediante un mecanismo de autenticación de usuarios.

- **Usabilidad:**

RFN3. El sistema debe usar un lenguaje y conceptos que sean familiares para el usuario, en lugar de términos informáticos o específicos del sistema.

RNF4. Los mensajes para interactuar con los usuarios y los de error deben ser lo suficientemente informativos, en idioma español y no deben revelar información interna.

- **Portabilidad:**

RNF5. El sistema debe ser ejecutado correctamente en cualquier plataforma y la mayoría de navegadores Web.

2.3.3. Historias de Usuario

Una historia de usuario es la unidad de trabajo más pequeña en un marco ágil. Es un objetivo final, no una función, expresado desde la perspectiva del usuario del software. Es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final o cliente. El propósito de una historia de usuario es articular cómo un elemento de trabajo entregará un valor particular al cliente. Son unas pocas frases en lenguaje sencillo que describen el resultado deseado. No entran en detalles, ya que los requisitos se añaden más tarde, una vez acordados por el equipo. En el SCRUM, las historias de los usuarios se añaden a los Sprints y se van realizando a lo largo del sprint. Es este trabajo sobre las historias de usuario lo que ayuda a los equipos de scrum a mejorar en la estimación y planificación de Sprints, lo que conduce a un pronóstico más preciso y a una mayor agilidad. (Rehkopf, 2023)

Para la descripción de requisitos funcionales de la propuesta de solución se define una historia de usuario para cada uno de ellos, para un total de 26 historias de usuario.

En el marco de la investigación fueron definidos los siguientes parámetros a tener en cuenta en las HU:

- **Número:** Número asignado a la HU.
- **Nombre:** Nombre de la HU.
- **Perfil (Yo como):** Rol del usuario final.
- **Necesidad (Quiero):** El objetivo que tiene la función de software para el usuario final.
- **Propósito (Para):** El objetivo de la experiencia del usuario final con la función de software.
- **Criterios de aceptación:** Aspectos importantes de interés para el usuario final.
- **Prototipo:** Diseño de la interfaz del sistema (opcional).

Tabla 2-2 Historia de Usuario: “HU01 - Autenticar usuario”

Fuente: Elaboración propia.

HU01 – Autenticar usuario	
Yo como:	Usuario del sistema
Quiero:	Autenticarme en el sistema
Para:	Tener acceso al sistema
Criterios de aceptación • Debe poseer un campo de Correo. • Debe poseer un campo de Contraseña. • Al presionar el botón Entrar se debe acceder a la aplicación web. • Al introducir credenciales incorrectas mostrar mensaje de “Usuario o contraseña incorrectos”. • En caso de que se quiera acceder a la aplicación web sin haber llenado los campos de Correo y Contraseña mostrar mensaje de “Este campo no puede estar vacío”. • En caso de que lo que se esté insertando no sea un correo mostrar mensaje de “El email no es válido” • En caso de que se esten insertando menos de 6 caracteres mostrar mensaje de “Este campo debe contener al menos 6 caracteres”.	

Prototipo:



Tabla 2-3 Historia de usuario: “HU07 - Editar Cliente”

Fuente: Elaboración propia.

HU07 – Editar Cliente	
Yo como:	Usuario del sistema
Quiero:	Editar un cliente en el sistema
Para:	Realizar modificaciones en los datos de los clientes registrados en el sistema
Criterios de aceptación Los datos a modificar se deben dividir en 3 secciones: General: • Código del cliente • Nombre oficial de la Entidad • Siglas • Fecha de creación • Organismo • OSDE/Grupo/Unión	

- Empresa
- Código REEUP u ONE
- NIT
- Registro Comercial
- Dirección
- Municipio
- Provincia

Contactos:

- Nombre y apellidos
- Carnet de identidad
- Email
- Teléfono

Datos bancarios

- Cuenta bancaria en CUP
- Nombre de la cuenta
- Entidad bancaria

Prototipo:

Gestión comercial

Cientes / **Editar cliente** ✓ Guardar

GENERAL CONTACTOS DATOS BANCARIOS

Código del cliente *	Nombre oficial de la entidad *	Siglas
115716	Sucursal Marlin Cayo Largo del Sur	Inserte las siglas
Fecha de creación	Organismo	OSDE / Grupo / Unión
dd / mm / aaaa	Inserte el organismo	Test place
Empresa	Código REEUP U ONE	NIT
Inserte la empresa	256.0.60655	30001501750
Registro comercial	Dirección	Municipio
Marinas	Cayo Largo del Sur, Los Canarreos	Inserte el municipio
Provincia	Inserte la provincia	

2.4 Planificación del Backlog

La Planificación del Backlog es una actividad que se realiza en el contexto de la metodología ágil Scrum para definir y priorizar los elementos de trabajo que se incluirán en el próximo Sprint. El objetivo de la Planificación del Backlog es asegurarse de que el equipo de desarrollo tenga una comprensión clara de los requisitos del cliente y de las funcionalidades que deben entregarse al final del Sprint (Radigan, 2023). Durante la Planificación del Backlog, el equipo de Scrum se reúne para discutir y definir los elementos de trabajo que se incluirán en el próximo Sprint. (Radigan, 2023). La Planificación del Backlog es una actividad importante en Scrum porque ayuda a asegurar que el equipo de desarrollo esté trabajando en las funcionalidades más importantes y valiosas para el cliente. También ayuda a garantizar que el equipo de desarrollo tenga una comprensión clara de lo que se espera de ellos durante el próximo Sprint (Radigan, 2023).

2.4.1 Product Backlog

El product backlog, también conocido como el trabajo pendiente de un producto, es una lista ordenada de todas las tareas que se harán para el desarrollo de un producto, especialmente cuando se utiliza la metodología Scrum. Esta lista se subdivide en backlogs o sprint backlogs, que son funciones o elementos que hay que finalizar y que forman parte de una hoja de ruta más amplia. El product backlog es esencialmente una lista de tareas pendientes específicas. Si el equipo aplica una metodología ágil y tiene en cuenta los principios del Manifiesto Ágil, el product backlog servirá para dividir proyectos y determinar cuáles son las tareas más importantes (Asana, 2022).

En términos más simples, un product backlog es una herramienta esencial para la gestión de proyectos que consiste en la elaboración de un listado de todas aquellas tareas que se quieren realizar durante el desarrollo de un proyecto con el objetivo de que estas sean visibles para todo el equipo. Esta lista debe ser ordenada y contar con una serie de elementos mediante los que se pueden desglosar y definir cada tarea de la forma más precisa posible, añadiendo en ella detalles como descripciones, orden de realización e importancia (Molina, 2023).

Es importante tener en cuenta que las tareas del product backlog no son definitivas e inamovibles y que no todos los elementos se podrán finalizar. Esto se debe a la naturaleza de las metodologías ágiles, que permiten ajustes y cambios a medida que los proyectos avanzan (Asana, 2022).

El Product Backlog contiene los siguientes campos:

- **ID:** Identificador único, simplemente un número autoincremental.
- **Nombre:** Descripción breve del requisito, normalmente de 2 a 10 palabras.

- **Importancia:** Ratio de importancia que el cliente da al requisito. Por ejemplo, 10. o 100. Más alto = más importante.
- **Estimación Inicial:** Valoración inicial del equipo referente a cuánto trabajo es necesario para implementar la funcionalidad, comparada con otras funcionalidades. La unidad es “puntos” y usualmente corresponde a “días x persona ideales”. Lo importante no es que las estimaciones absolutas sean correctas (es decir, que un requisito de 2 puntos deba durar 2 días), sino que las estimaciones relativas sean correctas (es decir, que un requisito de 2 puntos debería durar la mitad que una funcionalidad de 4 puntos).
- **Descripción:** Especificación de cómo funciona el requerimiento.

Tabla 2-4 Product Backlog.

Fuente: *Elaboración propia.*

ID	Nº	Nombre	Importancia	Estimación inicial	Descripción
1	RF 1	Autenticar usuario	70	2	Permite ingresar al usuario su correo y contraseña al sistema
2	RF2	Registrar usuario	70	2	Permitirá añadir un usuario en el sistema con sus datos correspondientes
3	RF 3	Editar usuario	70	2	Permitirá editar un usuario en el sistema
4	RF 4	Listar usuarios	70	2	Permitirá mostrar en forma de lista los usuarios en el sistema
5	RF 5	Deshabilitar usuario	70	2	Permitirá deshabilitar un usuario en el sistema
6	RF 6	Añadir cliente	100	2	Permitirá insertar un cliente en el sistema con sus datos correspondientes
7	RF 7	Editar cliente	90	2	Permitirá editar un cliente en el sistema
8	RF 8	Listar cliente	90	2	Permitirá mostrar en forma de lista los clientes en el sistema

9	RF 9	Añadir Tipo de Servicio	100	2	Permitirá insertar un Tipo de Servicio en el sistema con sus datos correspondientes.
10	RF 10	Editar Tipo de Servicio	90	2	Permitirá editar un Tipo de Servicio en el sistema
11	RF 11	Listar Tipo de Servicio	90	2	Permitirá mostrar en forma de lista los Tipos de Servicios en el sistema
12	RF 12	Añadir Servicio	100	6	Permitirá insertar un servicio en el sistema con sus datos correspondientes
13	RF 13	Editar Servicio	80	6	Permitirá editar un servicio en el sistema
14	RF 14	Listar Servicio	90	2	Permitirá mostrar en forma de lista los servicios en el sistema
15	RF 15	Añadir Solicitud de Servicio	100	2	Permitirá insertar una Solicitud de Servicio en el sistema con sus datos correspondientes
16	RF 16	Editar Solicitud de Servicio	90	2	Permitirá editar una Solicitud de Servicio en el sistema
17	RF 17	Listar Solicitud de Servicio	100	8	Permitirá mostrar en forma de lista las Solicitudes de Servicios en el sistema
18	RF 18	Crear Oferta Comercial	100	8	Permite crear una Oferta Comercial a partir de una Solicitud de Servicio añadida al sistema
19	RF 19	Editar Oferta Comercial	90	2	Permitirá editar una Oferta Comercial en el sistema
20	RF 20	Listar Oferta Comercial	90	2	Permitirá mostrar en forma de lista las Oferta Comerciales en el sistema
21	RF 21	Imprimir Oferta Comercial	80	2	Permitirá imprimir la Oferta Comercial

22	RF 22	Crear Contrato	100	2	Permite crear un Contrato en el sistema
23	RF 23	Listar Contrato	100	6	Permitirá mostrar en forma de lista los Contratos en el sistema
24	RF 24	Imprimir Contrato	90	2	Permitirá imprimir los contratos
25	RF 25	Crear Suplemento	100	4	Permite crear un Suplemento en el sistema
26	RF 26	Listar Suplemento	90	4	Permitirá mostrar en forma de lista los Suplementos en el sistema
27	RF 27	Añadir Reporte de Producción	100	8	Permite ingresar al sistema el Reporte de Producción de los servicios brindados en el mes a los clientes
28	RF 28	Editar Reporte de Producción	90	2	Permitirá editar el Reporte de Producción en el sistema
29	RF 29	Listar Reporte de Producción	90	2	Permitirá mostrar en forma de lista los Reportes de Producción
30	RF 30	Imprimir Reporte de Producción	90	2	Permitirá imprimir las facturas
31	RF 31	Imprimir Factura	100	2	Permitirá imprimir las facturas
32	RF32	Generar Contrato a imprimir	100	6	Permitirá generar el contrato para que sea impreso
33	RF33	Generar la Factura a imprimir	90	2	Permitirá generar la factura para que sea impresa, a partir del reporte de producción realizado

2.4.2 Sprint Backlog

El Sprint Backlog se compone del objetivo del Sprint (por qué), el conjunto de elementos del Product Backlog seleccionados para el Sprint (qué), así como un plan procesable para entregar el Incremento (cómo). El Sprint Backlog es un plan por y para los desarrolladores. Es una imagen muy visible y en tiempo real del trabajo que los desarrolladores planean realizar durante el Sprint para lograr el Objetivo del Sprint. En consecuencia, el Sprint Backlog se actualiza a lo largo del Sprint a medida que se aprende más. Debe tener suficientes detalles para que puedan inspeccionar su progreso en el *Daily Scrum*⁶ (Sutherland & Schwaber, 2020).

El Sprint Goal es el único objetivo del Sprint. Aunque el Sprint Goal es un compromiso de los Desarrolladores, proporciona flexibilidad en términos del trabajo exacto necesario para lograrlo. El Sprint Goal también crea coherencia y enfoque, animando al Scrum Team a trabajar juntos en lugar de en iniciativas separadas (Sutherland & Schwaber, 2020).

El objetivo del Sprint se crea durante el evento de planificación del Sprint y luego se agrega al Sprint Backlog. Mientras los desarrolladores trabajan durante el Sprint, tienen en cuenta el objetivo del Sprint. Si el trabajo resulta diferente a lo que esperaban, colaboran con el Product Owner para negociar el alcance del Sprint Backlog dentro del Sprint sin afectar el Sprint Goal (Sutherland & Schwaber, 2020). Para la planificación de las tareas se definieron un total de 3 Sprint que permiten el desarrollo completo de la herramienta.

⁶ El Daily SCRUM es una reunión diaria de máximo 15 minutos en la que los desarrolladores inspeccionan el avance desde la última reunión y planifican las siguientes 24 horas de trabajo. Esta reunión tiene como objetivo evaluar el progreso hacia la consecución del Objetivo del Sprint y adaptar el Sprint Backlog en consecuencia (Sutherland & Schwaber, 2020).

Tabla 2-5 Sprint backlog 1.
Fuente: Elaboración propia.

Sprint 1				
ID	Tarea	Asignada a	Estado	Tiempo
HU01-T1	Crear clases necesarias para gestionar el almacenamiento de usuario	Idalia González Brito	Terminada	2
HU01-T2	Crear los Endpoints necesarios para añadir usuario	Idalia González Brito	Terminada	2
HU01-T3	Diseñar vista para añadir usuario	Idalia González Brito	Terminada	4
HU01-T4	Crear clases necesarias para manipular los datos enviados y recibidos del backend para añadir usuario	Idalia González Brito	Terminada	4
HU02-T1	Crear los Endpoints necesarios para registrar usuario	Idalia González Brito	Terminada	2
HU02-T2	Diseñar vista para registrar usuario	Idalia González Brito	Terminada	2
HU03-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para registrar usuario	Idalia González Brito	Terminada	4
HU03-T1	Crear los Endpoints necesarios para editar usuario	Idalia González Brito	Terminada	2
HU03-T2	Diseñar vista para editar usuario	Idalia González Brito	Terminada	2

HU03-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para editar usuario	Idalia González Brito	Terminada	4
HU04-T1	Crear los Endpoints necesarios para listar usuario	Idalia González Brito	Terminada	2
HU04-T2	Diseñar vista para listar usuario	Idalia González Brito	Terminada	2
HU04-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para listar usuario	Idalia González Brito	Terminada	4
HU05-T1	Crear los Endpoints necesarios para deshabilitar usuario	Idalia González Brito	Terminada	2
HU05-T2	Diseñar vista para deshabilitar usuario	Idalia González Brito	Terminada	2
HU05-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para deshabilitar usuario	Idalia González Brito	Terminada	4
HU06-T1	Crear clases necesarias para gestionar el almacenamiento de clientes	Idalia González Brito	Terminada	2
HU06-T2	Crear los Endpoints necesarios para añadir clientes	Idalia González Brito	Terminada	2
HU06-T3	Diseñar vista para añadir clientes	Idalia González Brito	Terminada	4
HU06-T4	Crear clases necesarias para manipular los datos enviados y recibidos del backend para añadir clientes	Idalia González Brito	Terminada	4

HU07-T1	Crear los Endpoints necesarios para editar clientes	Idalia González Brito	Terminada	2
HU07-T2	Diseñar vista para editar clientes	Idalia González Brito	Terminada	4
HU07-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para editar clientes	Idalia González Brito	Terminada	4
HU08-T1	Crear los Endpoints necesarios para listar clientes	Idalia González Brito	Terminada	2
HU08-T2	Diseñar vista para listar clientes	Idalia González Brito	Terminada	4
HU08-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para listar clientes	Idalia González Brito	Terminada	4
HU09-T1	Crear clases necesarias para gestionar el almacenamiento de tipo de servicio	Idalia González Brito	Terminada	2
HU09-T2	Crear los Endpoints necesarios para añadir tipo de servicio	Idalia González Brito	Terminada	2
HU09-T3	Diseñar vista para añadir tipo de servicio	Idalia González Brito	Terminada	4
HU09-T4	Crear clases necesarias para manipular los datos enviados y recibidos del backend para añadir tipo de servicio	Idalia González Brito	Terminada	4
HU10-T1	Crear los Endpoints necesarios para editar un Tipo de Servicio	Idalia González Brito	Terminada	2

HU10-T2	Diseñar vista para editar un Tipo de Servicio	Idalia González Brito	Terminada	4
HU10-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para editar un Tipo de Servicio	Idalia González Brito	Terminada	4
HU11-T1	Crear los Endpoints necesarios para listar un Tipo de Servicio	Idalia González Brito	Terminada	2
HU11-T2	Diseñar vista para listar un Tipo de Servicio	Idalia González Brito	Terminada	4
HU11-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para listar un Tipo de Servicio	Idalia González Brito	Terminada	4
HU12-T1	Crear clases necesarias para gestionar el almacenamiento de servicio	Idalia González Brito	Terminada	2
HU12-T2	Crear los Endpoints necesarios para añadir servicio	Idalia González Brito	Terminada	2
HU12-T3	Diseñar vista para añadir servicio	Idalia González Brito	Terminada	4
HU12-T4	Crear clases necesarias para manipular los datos enviados y recibidos del backend para añadir servicio	Idalia González Brito	Terminada	4
HU13-T1	Crear los Endpoints necesarios para editar servicio	Idalia González Brito	Terminada	2
HU13-T2	Diseñar vista para editar servicio	Idalia González Brito	Terminada	2

HU13-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para editar servicio	Idalia González Brito	Terminada	4
HU14-T1	Crear los Endpoints necesarios para listar servicio	Idalia González Brito	Terminada	2
HU14-T2	Diseñar vista para listar servicio	Idalia González Brito	Terminada	4
HU14-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para listar servicio	Idalia González Brito	Terminada	4
HU15-T1	Crear clases necesarias para gestionar el almacenamiento de Solicitud de Servicio	Idalia González Brito	Terminada	2
HU15-T2	Crear los Endpoints necesarios para añadir Solicitud de Servicio	Idalia González Brito	Terminada	2
HU15-T3	Diseñar vista para añadir Solicitud de Servicio	Idalia González Brito	Terminada	4
HU15-T4	Crear clases necesarias para manipular los datos enviados y recibidos del backend para añadir Solicitud de Servicio	Idalia González Brito	Terminada	4
HU16-T1	Crear los Endpoints necesarios para editar Solicitud de Servicio	Idalia González Brito	Terminada	2
HU16-T2	Diseñar vista para editar Solicitud de Servicio	Idalia González Brito	Terminada	2
HU16-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para editar Solicitud de Servicio	Idalia González Brito	Terminada	4

HU17-T1	Crear los Endpoints necesarios para listar Solicitud de Servicio	Idalia González Brito	Terminada	2
HU17-T2	Diseñar vista para listar Solicitud de Servicio	Idalia González Brito	Terminada	4
HU17-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para listar Solicitud de Servicio	Idalia González Brito	Terminada	4

Tabla 2-6 Sprint backlog 2.
Fuente: Elaboración propia.

Sprint 2				
ID	Tarea	Asignada a	Estado	Tiempo
HU18-T1	Crear clases necesarias para gestionar el almacenamiento de Oferta Comercial	Idalia González Brito	Terminada	2
HU18-T2	Crear los Endpoints necesarios para añadir Oferta Comercial	Idalia González Brito	Terminada	2
HU18-T3	Diseñar vista para añadir Oferta Comercial	Idalia González Brito	Terminada	4
HU18-T4	Crear clases necesarias para manipular los datos enviados y recibidos del backend para añadir Oferta Comercial	Idalia González Brito	Terminada	4
HU19-T1	Crear los Endpoints necesarios para editar Oferta Comercial	Idalia González Brito	Terminada	2
HU19-T2	Diseñar vista para editar Oferta Comercial	Idalia González Brito	Terminada	2

HU19-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para editar Oferta Comercial	Idalia González Brito	Terminada	4
HU20-T1	Crear los Endpoints necesarios para listar Oferta Comercial	Idalia González Brito	Terminada	2
HU20-T2	Diseñar vista para listar Oferta Comercial	Idalia González Brito	Terminada	4
HU20-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para listar Oferta Comercial	Idalia González Brito	Terminada	4
HU21-T1	Implementar funcionalidad Imprimir Oferta Comercial	Idalia González Brito	Terminada	2
HU22-T1	Crear clases necesarias para gestionar el almacenamiento de Contrato	Idalia González Brito	Terminada	2
HU22-T2	Crear los Endpoints necesarios para añadir Contrato	Idalia González Brito	Terminada	2
HU22-T3	Diseñar vista para añadir Contrato	Idalia González Brito	Terminada	4
HU22-T4	Crear clases necesarias para manipular los datos enviados y recibidos del backend para añadir Contrato	Idalia González Brito	Terminada	4
HU23-T1	Crear los Endpoints necesarios para listar Contrato	Idalia González Brito	Terminada	2
HU23-T2	Diseñar vista para listar Contrato	Idalia González Brito	Terminada	4

HU23-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para listar Contrato	Idalia González Brito	Terminada	4
HU24-T1	Implementar funcionalidad Imprimir Contrato	Idalia González Brito	Terminada	2
HU25-T1	Crear clases necesarias para gestionar el almacenamiento de Suplemento	Idalia González Brito	Terminada	2
HU25-T2	Crear los Endpoints necesarios para añadir Suplemento	Idalia González Brito	Terminada	2
HU25-T3	Diseñar vista para añadir Suplemento	Idalia González Brito	Terminada	4
HU25-T4	Crear clases necesarias para manipular los datos enviados y recibidos del backend para añadir Suplemento	Idalia González Brito	Terminada	4
HU26-T1	Crear los Endpoints necesarios para listar Suplemento	Idalia González Brito	Terminada	2
HU26-T2	Diseñar vista para listar Suplemento	Idalia González Brito	Terminada	4
HU26-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para listar Suplemento	Idalia González Brito	Terminada	4

Tabla 2-7 Sprint backlog 3.
Fuente: Elaboración propia.

Sprint 3				
ID	Tarea	Asignada a	Estado	Tiempo
HU27-T1	Crear clases necesarias para gestionar el almacenamiento de Reporte de Producción	Idalia González Brito	Terminada	2
HU27-T2	Crear los Endpoints necesarios para añadir Reporte de Producción	Idalia González Brito	Terminada	2
HU27-T3	Diseñar vista para añadir Reporte de Producción	Idalia González Brito	Terminada	4
HU27-T4	Crear clases necesarias para manipular los datos enviados y recibidos del backend para añadir Reporte de Producción	Idalia González Brito	Terminada	4
HU28-T1	Crear los Endpoints necesarios para editar Reporte de Producción	Idalia González Brito	Terminada	2
HU28-T2	Diseñar vista para editar Reporte de Producción	Idalia González Brito	Terminada	2
HU28-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para editar Reporte de Producción	Idalia González Brito	Terminada	4
HU29-T1	Crear los Endpoints necesarios para listar Reporte de Producción	Idalia González Brito	Terminada	2
HU29-T2	Diseñar vista para listar Reporte de Producción	Idalia González Brito	Terminada	2

HU29-T3	Crear clases necesarias para manipular los datos enviados y recibidos del backend para listar Reporte de Producción	Idalia González Brito	Terminada	4
HU30-T1	Implementar funcionalidad Imprimir Reporte de Producción	Idalia González Brito	Terminada	2
HU31-T1	Implementar funcionalidad Imprimir Factura	Idalia González Brito	Terminada	2

2.5 Arquitectura de la solución

Según (Pressman & Maxim, 2021), en su forma más simple, la arquitectura del software es la estructura u organización de los componentes del programa, la manera en que estos interactúan y la estructura de datos que utilizan. El objetivo principal de la arquitectura del software es aportar elementos que ayuden a la toma de decisiones y, al mismo tiempo, proporcionar conceptos y un lenguaje común que permitan la comunicación entre los miembros de un proyecto. Para conseguirlo, la arquitectura del software construye abstracciones, materializadas en forma de diagramas comentados.

En este caso se determina el empleo del estilo “llamada y retorno”, lo cual permite obtener una estructura de programa fácil de modificar persiguiendo la escalabilidad del sistema. Según (Pressman & Maxim, 2021) el empleo de este estilo posibilita además la comunicación, la coordinación y la cooperación entre los componentes y las restricciones que definen cómo se integran para conformar el sistema, así como los modelos semánticos que facilitan al diseñador el entendimiento de todas las partes del sistema, evitando que las variaciones realizadas a funcionalidades o componentes específicos afecten el funcionamiento general.

2.5.1 Patrón arquitectónico Modelo – Vista – Controlador (MVC)

MVC es una propuesta de arquitectura del software utilizada para separar el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, lo que ofrece beneficios diversos.

MVC se usa inicialmente en sistemas donde se requiere el uso de interfaces de usuario, aunque en la práctica el mismo patrón de arquitectura se puede utilizar para distintos tipos de aplicaciones. Surge de

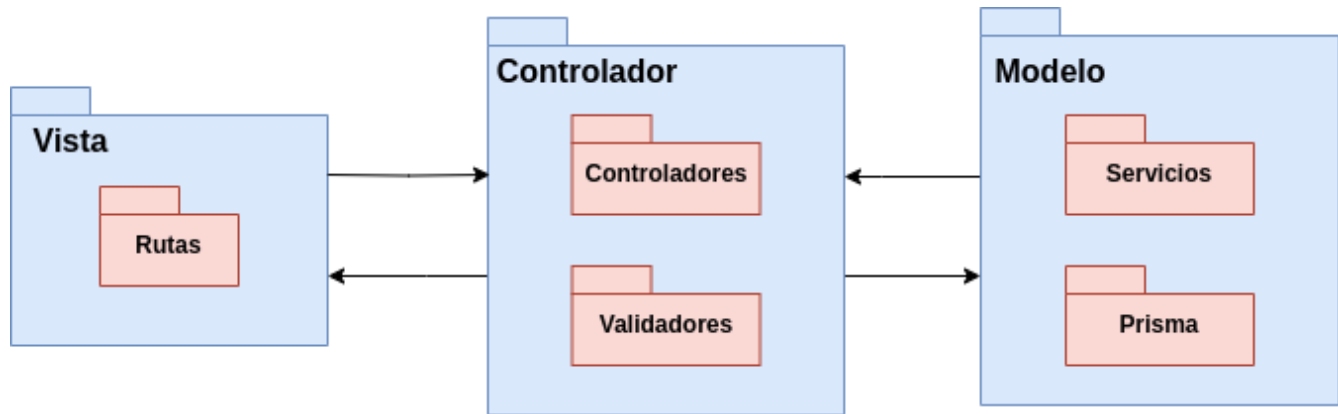
la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman **Modelos, Vistas y Controladores**, o lo que es lo mismo, *Model, Views & Controllers* (Alonso Aranda, 2019).

- **Modelo:** este componente se encarga de manipular, gestionar y actualizar los datos de una base de datos. No contiene ninguna lógica que describa cómo presentar los datos a un usuario.
- **Vista:** este componente presenta los datos del modelo al usuario. La vista sabe cómo acceder a los datos del modelo, pero no sabe que significa esta información o que puede hacer el usuario para manipularla.
- **Controlador:** este componente se encarga de gestionar las instrucciones que se reciben, atenderlas y procesarlas. Por medio del controlador se comunican el modelo y la vista: solicitando los datos necesarios, manipularlos para obtener los resultados y entregarlos a la vista para que pueda mostrarlos (Alonso Aranda, 2019).

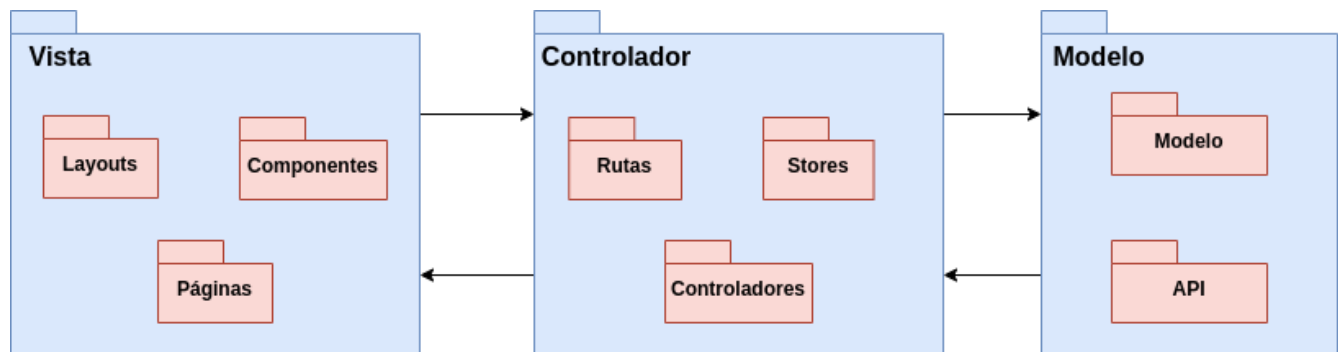
Modelo de la arquitectura del lado del servidor

*Figura 2-2 Modelo de la arquitectura del lado del servidor.
Fuente: Elaboración Propia.*



Modelo de la arquitectura del lado del cliente

*Figura 2-3 Modelo de la arquitectura del lado del cliente.
Fuente: Elaboración Propia.*



Conclusiones del capítulo

Con este capítulo, quedaron definidos los elementos esenciales para la construcción de la propuesta de solución. La realización del análisis y diseño del sistema a implementar permitió obtener una representación visual de las principales relaciones entre los conceptos asociados a la problemática. La generación de los artefactos de la metodología Scrum y la planificación de las tareas a cumplir mediante el Sprint Backlog. Se especificaron las funcionalidades y restricciones de la herramienta mediante la obtención de 33 requisitos funcionales y 5 requisitos no funcionales. Y se definió como arquitectura el patrón arquitectónico MVC.

CAPÍTULO III: Validación e implementación de la propuesta de solución.

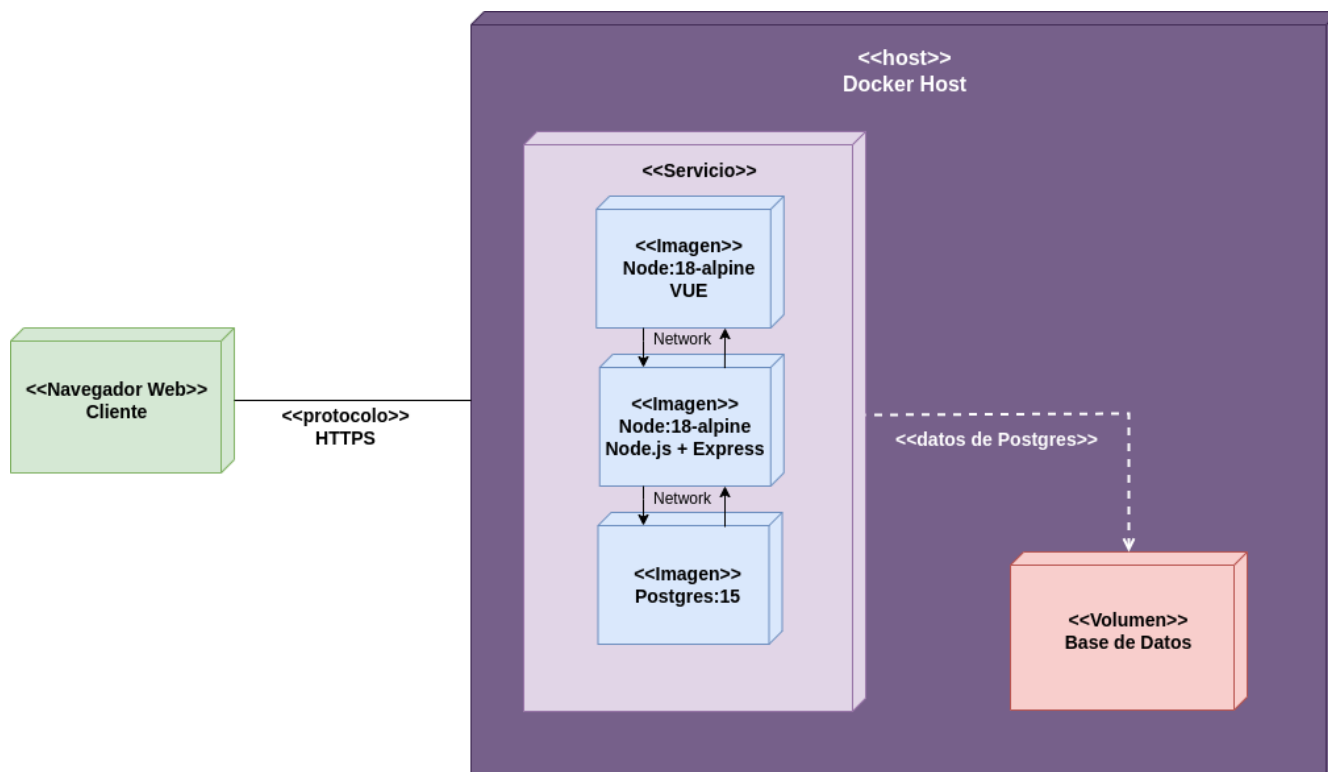
En el presente capítulo se exponen las especificaciones asociadas a la implementación de la aplicación. Se describen las pautas de codificación utilizadas, además la aplicación es sometida a un proceso de pruebas con el objetivo de verificar el cumplimiento de los requerimientos especificados anteriormente.

3.1 Fase de despliegue

Los diagramas de despliegue permiten modelar la disposición física o topología de un sistema; mostrando las relaciones entre sus componentes y las conexiones físicas entre el hardware. Un diagrama de despliegue está compuesto por: nodos, dispositivos y conectores (Sommerville, 2019).

Figura 3-1 Diagrama de despliegue.

Fuente: Elaboración propia.



En la Figura 3-1, se muestra el diagrama de despliegue para la propuesta de solución. El despliegue del sistema consiste en un servidor (host) que ejecuta el servicio de Docker⁷ y mantiene en ejecución

⁷ Docker es un proyecto de código abierto que automatiza la implementación de aplicaciones como contenedores portátiles y autosuficientes que se pueden ejecutar en la nube o localmente (Kane & Matthias, 2018).

3 contenedores, uno para el sistema del lado del cliente, otro para el sistema del lado del servidor y el último que contiene un servidor de Postgres para gestionar los datos de forma permanente. Además, para el contenedor que ejecuta Postgres se cuenta con un volumen de información donde será almacenada en el disco la base de datos. Dicho sistema será accedido mediante un navegador web por sus usuarios.

3.2 Estándares de codificación

Los estándares de codificación son un conjunto de pautas, mejores prácticas, estilos de programación y convenciones que los desarrolladores siguen cuando escriben código fuente para un proyecto. Los desarrolladores deben seguir estos estándares para, entre otras cosas:

- Mantener el código mantenible
- Mantener el código transparente, sano y legible
- Mantener el código escalable.

Cuando un proyecto se rige por una guía de estilo, esta debe aprenderse, entenderse e implementarse en todo momento; cualquier desviación debe estar plenamente justificada (University of St Andrews, 2022).

Sangría:

- Se debe utilizar 2 espacios para la sangría.

Grosor de línea:

- Cada línea de código no debe exceder los 80 caracteres en la medida de lo posible (en circunstancias especiales, puede exceder ligeramente los 80, pero la más larga no puede exceder los 120).

Comillas:

- Todas las comillas deben ser simples '...'.

Final de Línea:

- En JavaScript el punto y coma (;) al final de cada línea es opcional, se debe utilizar para mejor legibilidad y delimitación.

Figura 3-2 Ejemplo de estándares de codificación.

Fuente: Elaboración propia.

```

12  async function fetch() : Promise<void> {
13
14      try {
15          loading.value = true;
16          const response : GetClientLabelsResponse = await makeClientApi().getClientLabels();
17          items.value = response.data;
18      } catch (e) {
19          console.log(e);
20          snackbar.push({
21              color: 'error',
22              text: 'Error cargando los clientes.',
23          });
24      } finally {
25          loading.value = false;
26      }
27  }
28  return { items, fetch, loading };
29  }

```

3.2 Pruebas de software

Las pruebas de software son actividades que garantizan que el sistema que se está desarrollando cumpla con los estándares de calidad requeridos (Pressman & Maxim, 2019). Otra ventaja importante de las pruebas de software es que permiten al equipo de desarrollo detectar y corregir la mayoría de las fallas, desperfectos y errores del software antes de que este sea entregado al cliente (Sommerville, 2019). Esto significa que las pruebas deben estar presentes en cualquier proceso de software cuyo objetivo sea entregar un producto de calidad, según lo que se espera de él. En resumen, entre los objetivos más notables de las pruebas y correcciones del sistema se pueden mencionar:

- Comprobar si el sistema implementa las funcionalidades predefinidas por el cliente.
- Proporcionar al equipo del proyecto información sobre las funcionalidades del sistema que no funcionan según lo previsto.
- Corregir a tiempo los defectos, errores, fallas e inconformidades encontrados.

3.2.1 Estrategia de prueba

Una estrategia de prueba del software integra los métodos de diseño de caso de pruebas del software en una serie bien planeada de pasos que desembocará en la eficaz construcción del mismo. La estrategia proporciona un mapa que describe los pasos que se darán como parte de la prueba, indica cuándo se planean, cuándo se dan estos pasos, además de cuanto esfuerzo, tiempo y recursos consumirá. Por tanto, cualquier estrategia de prueba debe incorporar la planeación de pruebas, el

diseño de casos de pruebas, la ejecución de pruebas, la recolección y evaluación de los datos resultantes (Pressman & Maxim, 2021).

La estrategia de prueba debe indicar los niveles de pruebas (ciclos) que se aplican y la intensidad o profundidad a aplicar para cada nivel de prueba definido.

3.2.2 Niveles de prueba

La prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Se distinguen los siguientes Niveles de Pruebas (Jacobson et al., 2000).

- **Prueba de Unidad:** Un componente es la unidad más pequeña especificada de un sistema, las pruebas se llevan a cabo tras la construcción o realización de cada componente para verificar que la implementación se esté llevando conforme a los estándares acordados. El objetivo es comprobar que el sistema, entendido como una unidad funcional, está correctamente codificado.
- **Prueba de Integración:** Son las pruebas que se hacen cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados.
- **Prueba de Aceptación:** Es la prueba final antes del despliegue del sistema. El propósito es confirmar que el sistema está terminado, que desarrolla puntualmente las necesidades de la organización y que es aceptado por los usuarios finales.

A continuación, se ofrecen más detalles sobre las pruebas realizadas.

3.2.3 Pruebas unitarias

Se centra en el esfuerzo de verificación de la unidad más pequeña del diseño del software, el componente o sistema de software. Tomando como guía la descripción del diseño al nivel de componentes, se prueban importantes caminos de control para describir errores dentro de los límites del módulo. El alcance restringido que se ha determinado para las pruebas de unidad limita la relativa complejidad de las pruebas y los errores que estas descubren. Las pruebas se encuentran en la lógica del procedimiento interno y en las estructuras de datos dentro de los límites de un componente. Este tipo de prueba se puede aplicar en paralelo a varios componentes (Pressman & Maxim, 2021).

Para la realización de las pruebas unitarias en la propuesta de solución se hizo uso de Jest.js, un marco de prueba de JavaScript diseñado para garantizar la exactitud de cualquier código base de JavaScript. Jest permite escribir pruebas con una API accesible, familiar y rica en funciones que brinda resultados

rápidamente (jestjs, 2017). A continuación, se muestran en las Figura 3-3 los resultados de las pruebas unitarias realizadas a los diferentes CRUD⁸ del sistema.

Figura 3-3 Pruebas unitarias al sistema.

Fuente: Elaboración propia.

```
> server@0.1.0 test
> jest
PASS src/__test__/index.test.ts (6.144 s)
  Test endpoints
    Test client CRUD
      ✓ add client (79 ms)
      ✓ edit client (165 ms)
      ✓ get client (101 ms)
      ✓ get all clients (107 ms)
    Test service CRUD
      ✓ add service (56 ms)
      ✓ edit service (138 ms)
      ✓ get service (110 ms)
      ✓ get all services (167 ms)
    Test service type CRUD
      ✓ add service type (99 ms)
      ✓ edit service type (178 ms)
      ✓ get service type (83 ms)
      ✓ get all service types (149 ms)
    Test service request CRUD
      ✓ add service request (148 ms)
      ✓ edit service request (127 ms)
      ✓ get service request (87 ms)
      ✓ get all service requests (161 ms)
    Test commercial offer CRUD
      ✓ add commercial offer (148 ms)
      ✓ edit commercial offer (164 ms)
      ✓ get commercial offer (85 ms)
      ✓ get all commercial offers (159 ms)
    Test contract CRUD
      ✓ add contract (69 ms)
      ✓ edit contract (129 ms)
      ✓ get contract (124 ms)
      ✓ get all contracts (149 ms)
    Test production report CRUD
      ✓ add production report (69 ms)
      ✓ edit production report (122 ms)
      ✓ get production report (89 ms)
      ✓ get all production reports (143 ms)
    Test user CRUD
      ✓ add user (88 ms)
      ✓ edit user (129 ms)
      ✓ get user (84 ms)
      ✓ get all users (141 ms)
    Test generating documents for printing
      ✓ generate commercial offer PDF (76 ms)
      ✓ generate contract PDF (140 ms)
      ✓ generate production report PDF (84 ms)
      ✓ generate invoice PDF (121 ms)
  Test Suites: 1 passed, 1 total
  Tests: 36 passed, 36 total
  Snapshots: 0 total
  Time: 6.275 s
  Ran all test suites.
```

3.2.5 Pruebas de aceptación

Cuando se construye un software para un cliente, se realizan una serie de pruebas de aceptación a fin de permitir al cliente validar todos los requerimientos. 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 período de semanas o meses y mediante ella descubrir errores acumulados que con el tiempo puedan degradar el sistema. La mayoría de los constructores de productos de software usan un proceso llamado prueba alfa y prueba beta para descubrir errores que al parecer solo el usuario final es capaz de encontrar (Pressman & Maxim, 2021).

⁸ CRUD: Create (Crear), Read (Leer), Update (Actualizar) y Delete (Borrar) es un acrónimo para las maneras en las que se puede operar sobre información almacenada. Es un mnemónico para las cuatro funciones del almacenamiento persistente. CRUD usualmente se refiere a operaciones llevadas a cabo en una base de datos o un almacén de datos, pero también puede aplicar a funciones de un nivel superior de una aplicación como eliminado suave donde la información no es realmente eliminada, sino es marcada como eliminada a través de un estatus (IEEE Staff, 2021).

Tabla 3-1 Prueba de Aceptación.
Fuente: Elaboración propia.

ID	PA1
Historia de usuario	HU1 - Autenticar Usuario
Fecha de escritura	
Fecha de ejecución	
Aprobado / ID del defecto:	Aprobado
Descripción • La interfaz de la autenticación debe mostrarse como en el prototipo. • El usuario debe introducir sus credenciales. • Al presionar el botón “Entrar” se deben validar los datos insertados. • Al introducir credenciales incorrectas el sistema debe mostrar el mensaje de “Usuario o contraseña incorrectos”. • Al dejar campos en blanco el sistema debe mostrar el mensaje de “Este campo no puede estar vacío” debajo de los respectivos campos en blanco. • Al insertar un correo no válido el sistema debe mostrar el mensaje de “El correo no es válido”. • En caso de que se estén insertando menos de 6 caracteres en la contraseña el sistema debe mostrar el mensaje de “Este campo debe contener al menos 6 caracteres”. • Si no existe ningún problema al presionar el botón “Entrar” el usuario debe poder entrar al sistema.	
Resultado esperado 1. El usuario debe poder acceder al sistema de forma satisfactoria insertando su correo y su contraseña.	

En los anexos se encuentra el resto de las pruebas de aceptación que se realizaron

3.2.6 Pruebas de integración

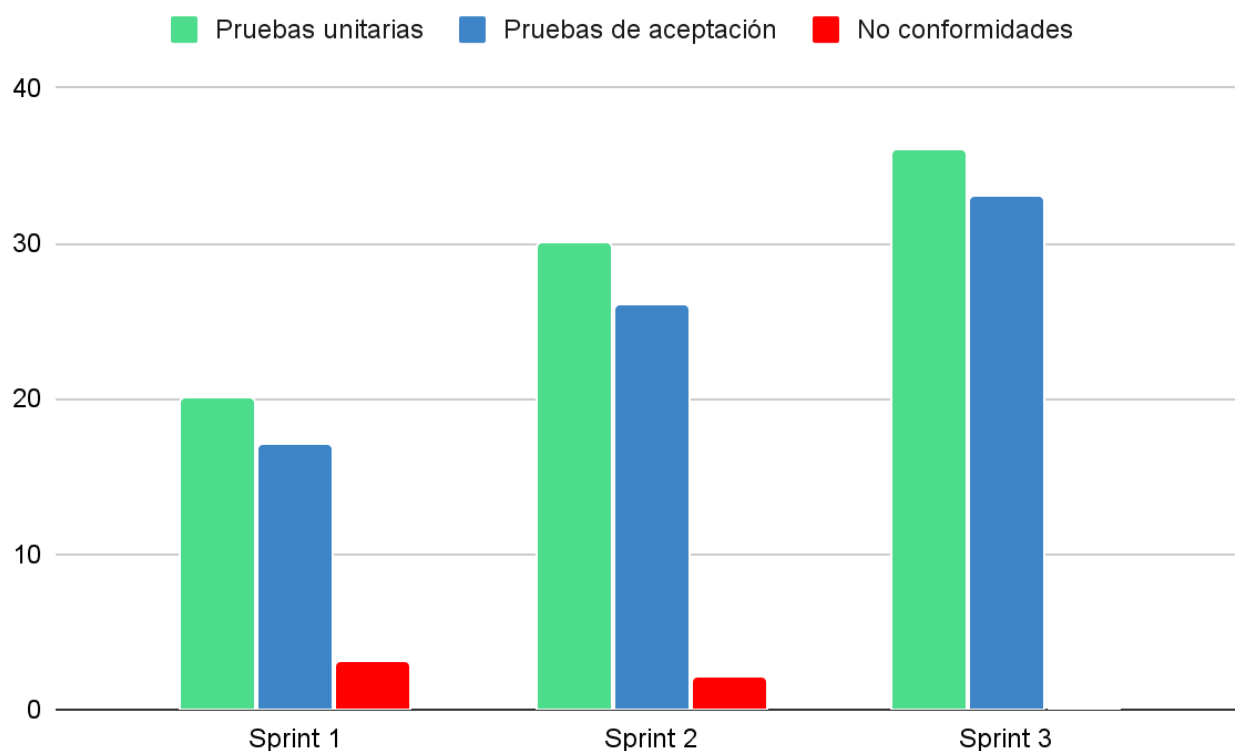
Las pruebas del sistema durante el desarrollo incluyen la integración de componentes para crear una versión del sistema y, luego, poner a prueba el sistema integrado. Las pruebas de sistema demuestran que los componentes son compatibles, que interactúan correctamente y que transfieren los datos correctos en el momento adecuado a través de sus interfaces (Sommerville, 2019).

En el caso de la propuesta solución se realizaron pruebas de integración entre el servidor que le presta servicios a la aplicación web y viceversa, inicialmente se utilizó la herramienta Postman para comprobar que la api rest tenga un correcto funcionamiento. Posteriormente se realizó una prueba general a todas las funcionalidades de la aplicación arrojando resultados satisfactorios.

3.2.7 Resultado de las pruebas aplicadas

En la Figura 3-4, se muestran los resultados de las pruebas aplicadas, donde se ejecutaron un total de 3 iteraciones. En el siguiente gráfico de barras se evidencia, además, el total de no conformidades identificadas por cada iteración.

Figura 3-4 Resultado de las pruebas aplicadas.
Fuente: Elaboración Propia.



En la primera iteración se implementan las funcionalidades descritas en 17 historias de usuario, se ejecutaron 17 pruebas de aceptación y 20 pruebas unitarias, encontrando así 3 no conformidades. Todas las no conformidades de la primera iteración fueron subsanadas en la segunda. Una vez corregidos los problemas detectados, se implementan las funcionalidades de 9 historias de usuario. Se repitieron las pruebas anteriores y se aplicaron otras 9 pruebas de aceptación y 10 pruebas unitarias

con el objetivo de validar las nuevas funcionalidades encontrando 2 no conformidades. En la tercera iteración se corrigieron las no conformidades detectadas y se implementan las funcionalidades de las 7 historias de usuario restantes, se realizaron otras 6 pruebas unitarias y 7 pruebas de aceptación y no se encontraron no conformidades, validándose de esta manera el correcto funcionamiento de la herramienta propuesta. De un total de 69 pruebas realizadas durante todo el proceso de desarrollo, solo 5 resultaron no satisfactorias. A continuación, en la tabla 3-2 se ofrecen detalles de las no conformidades detectadas. Finalmente, se repitieron todas las pruebas resultando satisfactorias.

Tabla 3-2 No conformidades.

Fuente: Elaboración propia.

Sprint	Tipo de prueba	No conformidad
1	Prueba de aceptación PA2	El usuario guardado se guarda por defecto con el estado Inactivo por lo que no permite que este se autentique en el sistema sin modificar su estado antes a Activo.
1	Prueba de aceptación PA2	Falta el campo de dirección.
1	Prueba de aceptación PA3	Falta el campo de dirección.
2	Prueba de aceptación PA23	El estado del contrato no es mostrado, no se puede determinar si cada elemento es una proforma de contrato o un contrato firmado.
2	Prueba unitaria Generar PDF Contrato	El PDF del contrato no es debidamente generado presentando errores al mostrarlo.

Conclusiones del capítulo

En este capítulo se diseñó el diagrama de despliegue para facilitar la comprensión de la distribución del software en el ambiente de producción. Se definieron estándares de codificación para garantizar que otras personas puedan entender el código y depurar partes del mismo. Se emplearon las pruebas unitarias, de aceptación e integración para comprobar el correcto flujo de trabajo de la solución, así como identificar y corregir las no conformidades detectadas en cada iteración, garantizando así el correcto funcionamiento de la herramienta.

CONCLUSIONES FINALES

Durante el desarrollo de la presente investigación, descrita en el documento, se ha dado cumplimiento a los objetivos planteados, construyendo una herramienta de gestión de contratación y facturación para la UEB-Datos de la Empresa de Servicios Especializados GET, que favorece la ejecución de tareas de manera correcta, fácil e intuitiva y beneficiando a los profesionales y especialistas de la institución, incrementando la productividad del personal.

Entre las principales conclusiones a las que se puede arribar con la investigación y trabajo desarrollado se encuentran:

- La definición de los fundamentos teóricos estableció las bases de conocimiento necesarias para comprender los conceptos básicos de la gestión de la contratación y la facturación, lo que facilitó en etapas posteriores la captura de requisitos en función de que la aplicación web contará con las características necesarias para cumplir el objetivo general.
- El estudio correspondiente de sistemas homólogos, las herramientas y tecnologías permitió que su elección fuese la más adecuada para la correcta y rápida implementación del sistema.
- Con el diseño e implementación de las funcionalidades propuestas se obtuvo una aplicación informática que responde a las necesidades del cliente definidas durante la etapa de análisis.
- El cumplimiento de las actividades propuestas por la metodología SCRUM para la implementación de la aplicación web, permitió que se generaran los artefactos necesarios para la construcción del software de manera organizada, tal fue el caso de la especificación de requisitos, las historias de usuario, la planificación del Backlog, los Sprints Backlog y el patrón arquitectónico.
- El desarrollo de las diferentes pruebas de software permitió detectar defectos, errores y fallas que pudieron ser corregidos a tiempo, logrando de esta manera que la aplicación para gestión de contratación y facturación para la UEB-Datos de la Empresa de Servicios Especializados GET tuviera la calidad necesaria y fuera de utilidad según las peticiones del cliente.

RECOMENDACIONES

Para dar continuidad a la presente investigación se recomienda incorporar al sistema un módulo de Registro de Cobro a través del cual se pueda llevar la gestión de documentos bancarios como son los pagos de los clientes y los medios que estos usan para efectuar el pago ejemplo cheque o transferencia y en dicho módulo gestionar un registro de la liquidación de las facturas.

REFERENCIAS BIBLIOGRÁFICAS

A. Cobo, N. (2019, March 18). Empresas y Empresarios en Cuba.

<https://horizontecubano.law.columbia.edu/sites/default/files/content/docs/NCR%20Empresas%20I%20%28ed%29%20%2B%20Access.pdf>

Ahora Freeware ERP sin coste de licencias. (n.d.). Labelgrup.

<https://www.labelgrup.com/ahora-freeware>

Alonso Aranda, C. (2019, June 7). *CENTRO DE ESTUDIOS POSTGRADO*. Universidad de Jaén.

https://crea.ujaen.es/bitstream/10953.1/11437/1/ALONSO_ARANDA_CARLOS_TFM_INFORMATICA.pdf

Angular. (2023). *Angular - Introducción a los conceptos de Angular*. Angular.

<https://docs.angular.lat/guide/architecture>

Aprendiendo Arduino. (2017, July 4). *Qué es la Programación | Aprendiendo Arduino*.

Aprendiendo Arduino. <https://aprendiendoarduino.wordpress.com/2017/07/04/que-es-la-programacion/>

Asana. (2022, August 25). *Qué es el product backlog y guía para hacer uno con ejemplo [2022] • Asana*. Asana. <https://asana.com/es/resources/product-backlog>

Beck, K. (2018, June). *Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software*. Catálogo de recursos SCALA.

<https://gc.scalahed.com/recursos/files/r161r/w25597w/438760423-269-823-1-PB-pdf.pdf>

Canizales Muñoz, L. D. (2020, June 30). *Elementos clave de la innovación empresarial. Una revisión desde las tendencias contemporáneas*. REVISTA INNOVA ITFIP.

<https://doi.org/10.54198/innova06.03>

- Cuba. Ministerio de Justicia. (2013). *Derecho-ley no. 304 "De la contratación económica"*. El Ministerio.
- Darias, S. (2023, February 16). *¿Qué es y para qué sirve un TDD o Test Driven Development?* Intelequia. <https://intelequia.com/es/blog/post/qu%C3%A9-es-y-para-qu%C3%A9-sirve-un-tdd-o-test-driven-development>
- Drumond, C. (2023). *Scrum: qué es, cómo funciona y cómo empezar*. Atlassian. <https://www.atlassian.com/es/agile/scrum>
- El software de gestión comercial más completo*. (n.d.). OfiGes: Software de Gestión Comercial | Software Facturación. <https://www.ofiges.info/>
- Figueroa Morán, G. L. (2017). *MODELO DE PLAN ESTRATÉGICO DE SISTEMAS PARA LA GESTIÓN Y ORGANIZACIÓN A TRAVÉS DE UNA PLATAFORMA INFORMÁTICA*. 3Ciencias.
- Frederick, D. (2023, enero). *¿Qué es una factura? - Características, elementos y tipos*. Enciclopedia Económica. <https://enciclopediaeconomica.com/factura/>
- Gandarillas, A. (2017, JUNIO 25). *Marco de trabajo*. METODOLOGÍA Metodologías para el desarrollo y mantenimiento de software y sistemas de información. <https://metodologia.es/marco-de-trabajo/>
- Gil, S. (2016, February 6). *Contrato - Qué es, definición y concepto | 2023*. Economipedia. <https://economipedia.com/definiciones/contrato.html>
- Git SCM. (2023). Git SCM. <https://git-scm.com/>
- Green, D., & Green, P. O. D. o. P. D. (2016). *Express. Js: Node. Js Framework for Web Application Development*. CreateSpace Independent Publishing Platform.
- Heddings, A. (2020, August 26). *What Is Node.JS and What Is It Used for?* HowToGeek. <https://www.howtogeek.com/devops/what-is-node-js-and-what-is-it-used-for/>

IBM. (2023, enero 13). *Sentencia ROLLBACK*. IBM.

<https://www.ibm.com/docs/es/db2/11.5?topic=statements-rollback>

IEEE Staff. (2021). *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems Technology and Applications (IDAACS)*. IEEE.

Introducción — Vue.js. (n.d.). Vue.js. <https://es.vuejs.org/v2/guide/>

Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software*. Pearson Educación.

jestjs. (2017). *jestjs*. Jest · Delightful JavaScript Testing. <https://jestjs.io/>

Jiménez López, R. (2003). *Análisis y Diseño Orientado a Objetos de un Framework para el Modelado Estadístico con MLG*. TDX.

<https://www.tdx.cat/bitstream/handle/10803/9442/trj11de1.pdf;jsessionid=3C568DA874A2DF37CCA6347E3F54C06E?sequence=1>

Johnson, B. (2019). *Visual Studio Code: End-To-End Editing and Debugging Tools for Web Developers*. Wiley.

Kane, S. P., & Matthias, K. (2018). *Docker: Up & Running: Shipping Reliable Containers in Production*. O'Reilly Media, Incorporated.

Lago, N. (2022, June 16). *Marcos de trabajo o framework en diferentes etapas del desarrollo de software*. Saasradar. <https://saasradar.net/marcos-de-trabajo-o-framework/>

Larman, C. (2004). *Applying UML and Patterns*. Prentice Hall PTR.

MacNeil, C. (2022, November 9). *Cómo elaborar un proceso de contratación de primer nivel [2022] • Asana*. Asana. <https://asana.com/es/resources/hiring-process>

Mammadli, E., & Klivak, V. (2020). *Measuring the Effect of the Digitalization*. University of Tartu, Faculty of Economics and Business Administration.

Meloni, J. C., & Kyrnin, J. (2019). *Sams Teach Yourself HTML, CSS, and JavaScript All in One*. Pearson.

- Metodologías ágiles - ¿Qué son?, ventajas, desventajas y las más usadas.* (2017 - 2023). Enciclopedia Económica. <https://enciclopediaeconomica.com/metodologias-agiles/>
- Milformatos. (2022, julio 22). *Reporte de Producción > Ejemplos y Formatos» Descarga Gratis.* Milformatos.com. <https://milformatos.com/empresas-y-negocios/reporte-de-produccion/>
- Molina, D. (2023, June 27). *Qué es un product backlog y cómo hacer uno [Guía Scrum].* IEBS. <https://www.iebschool.com/blog/que-es-un-product-backlog-y-como-hacer-uno-guia-scrum-agile-scrum/>
- Morales, G. (2019). *Introducción a Node.js.* Independently Published.
- Moreno Pérez, J. C. (2018). *Entornos de desarrollo.* Editorial Síntesis.
- Muñoz, F. (2023, February 1). *Los 12 mejores programas de facturación gratis en 2023.* Holded. <https://www.holded.com/es/blog/los-mejores-programas-de-facturacion>
- Núñez, A. (2021, May 11). *¿Qué es una API y para qué sirve? Todo lo que necesitas saber.* Business Insider España. <https://www.businessinsider.es/api-sirve-todo-necesitas-saber-861403>
- Orellana, P. (2019, December 30). *Gestión de procesos - Qué es, definición y concepto | 2023.* Economipedia. <https://economipedia.com/definiciones/gestion-de-procesos.html>
- Pascual, J. R. (2019, October 22). *Capas y niveles: Diseño y confusión.* Disrupción Tecnológica. <https://www.disrupciontecnologica.com/capas-y-niveles-diseno-y-confusion/>
- Patrones de Arquitectura – Estilos de Arquitectura (Cap. 3) – RJ Code Advance.* (n.d.). RJ Code Advance. <https://rjcodeadvance.com/patrones-de-software-que-es-patron-de-arquitectura-parte-3/>
- PMOInformatica. (2012, September 3). *Ambientes de desarrollo de software: Buenas prácticas - La Oficina de Proyectos de Informática.* PMOInformatica. <http://www.pmoinformatica.com/2012/09/ambientes-de-desarrollo-de-software.html>

- Postgres team. (2023). *Documentation: 16: 1. What Is PostgreSQL?* PostgreSQL.
<https://www.postgresql.org/docs/current/intro-what-is.html>
- Pressman, R. S., & Maxim, B. R. (2019). *Loose Leaf for Software Engineering: A Practitioner's Approach*. McGraw-Hill Education.
- Pressman, R. S., & Maxim, B. R. (2021). *Engenharia de software* (Ferraretto E Costa Ltda., Trans.). AMGH.
- Productos. (n.d.). Productos | Desoft. <https://www.desoft.cu/es/productos/322>
- Qué es Node.js y por qué debería usarlo. (2023, June 15). Kinsta. <https://kinsta.com/es/base-de-conocimiento/que-es-node-js/>
- ¿Qué es una “startup”? (2022, April 21). ¿Qué es una “startup”?
<https://www.santander.com/es/stories/que-es-una-startup>
- ¿Qué son los mnemónicos? | Indiantown Middle School. (n.d.). Martin County School District.
<https://www.martinschools.org/o/ims/page/son-los-mnemicos>
- Quipu. (2023). *Software de facturación para empresas*. Quipu.
<https://getquipu.com/es/empresas>
- Radigan, D. (2023). *Explicación del backlog del producto [+ ejemplos]*. Atlassian.
<https://www.atlassian.com/es/agile/scrum/backlogs>
- React. (2023). *Quick Start – React*. React. <https://react.dev/learn>
- Rehkopf, M. (2023). *Historias de usuario | Ejemplos y plantilla*. Atlassian.
<https://www.atlassian.com/es/agile/project-management/user-stories>
- Sommerville, I. (2019). *Engineering Software Products: An Introduction to Modern Software Engineering*. Pearson.
- Startechup. (2021, January 15). *Qué, cuándo y dónde utilizarlo*. Startechup.
<https://www.startechup.com/es/blog/express-js-what-it-is-used-for-and-when-where-to-use-it-for-your-enterprise-app-development/>

Stefaniak, P. (2019, July 26). *¿Qué es Backend y Frontend?* Descubre Comunicación.

<https://descubrecomunicacion.com/que-es-backend-y-frontend/>

Sutherland, J., & Schwaber, K. (2020). *The 2020 Scrum Guide TM*. Scrum Guides.

<https://scrumguides.org/scrum-guide.html>

TechTerms.com. (2020, April 11). *Backend Definition*. TechTerms.com.

<https://techterms.com/definition/backend>

University of St Andrews. (2022). *Code standards - Digital standards*. University of St Andrews.

<https://www.st-andrews.ac.uk/digital-standards/code-standards/>

Villaruel, J. (2016, July 19). *ACCIÓN PROFORMA, OTORGAMIENTO Y FIRMA*. abogado en consulta jurídica. <http://tuabogadoenvivo.com/accion-proforma-otorgamiento-firma-escritura-publica/>

Vuetify 2. (2023, September 29). *Why you should be using Vuetify*. Vuetify 2.

<https://v2.vuetifyjs.com/en/introduction/why-vuetify/#why-vuetify3f>

Xetid. (2021, September 22). *Distra*. <https://www.xetid.cu/es/productos/distra>

ANEXOS

Anexo 1: Historias de usuario

Tabla A1-1 Historia de usuario “HU02 - Registrar usuario”

Fuente: Elaboración propia.

HU02 – Registrar usuario	
Yo como:	Administrador del sistema
Quiero:	Registrar un usuario en el sistema
Para:	Que pueda tener acceso al sistema
Criterios de aceptación • El campo de nombre es requerido y debe señalar un error en caso de dejarlo vacío. • El campo de apellidos es requerido y debe señalar un error en caso de dejarlo vacío. • El campo de correo es requerido y debe señalar un error en caso de dejarlo vacío. • El campo estado debe indicar si el usuario está “Activo”, “Inactivo” o si está “Eliminado”. • El campo de teléfono es opcional. • El campo de dirección es opcional. • El campo de contraseña es requerido y debe señalar un error en caso de dejarlo vacío o que contenga menos de 6 caracteres. • Tener la opción “Guardar”.	

Tabla A1-2 Historia de usuario “HU03 - Editar usuario”

Fuente: Elaboración propia.

HU03 – Editar usuario	
Yo como:	Administrador del sistema
Quiero:	Editar los datos de un usuario
Para:	Modificar los datos de ese usuario en el sistema

Criterios de aceptación

- El campo de nombre es requerido y debe señalar un error en caso de dejarlo vacío.
- El campo de apellidos es requerido y debe señalar un error en caso de dejarlo vacío.
- El campo de correo es requerido y debe señalar un error en caso de dejarlo vacío.
- El campo estado debe indicar si el usuario está “Activo”, “Inactivo” o si está “Eliminado”.
- El campo de teléfono es opcional.
- El campo de dirección es opcional.
- El campo de contraseña es requerido y debe señalar un error en caso de dejarlo vacío o que contenga menos de 6 caracteres.
- Tener la opción “Guardar”.

Tabla A1-3 Historia de usuario “HU04 - Listar usuario”**Fuente: Elaboración propia.**

HU04 – Listar usuario	
Yo como:	Administrador del sistema
Quiero:	Ver una lista con todos los usuarios
Para:	Ver una breve descripción de los usuarios del sistema
Criterios de aceptación • Ver en forma de lista a los usuarios del sistema. • Tener la opción de “Editar” por cada usuario de la lista.	

Tabla A1-4 Historia de usuario “HU05 – Deshabilitar Usuario”**Fuente: Elaboración propia.**

HU05 – Deshabilitar Usuario	
Yo como:	Administrador del sistema
Quiero:	Deshabilitar un Usuario del sistema
Para:	Que el usuario no tenga acceso al sistema

Criterios de aceptación

- En el botón “Editar usuario” en el campo “Estado” seleccionar “Eliminado”.

Tabla A1-5 Historia de usuario “HU06 – Añadir Cliente”

Fuente: Elaboración propia.

HU06 – Añadir Cliente	
Yo como:	Usuario del sistema
Quiero:	Añadir un Cliente en el sistema
Para:	Poder tener acceso a su información personal en el sistema
Criterios de aceptación • Llenar los campos con los datos personales del Cliente, priorizando los campos que son obligatorios. Los datos del Cliente a completar se deben dividir en 3 secciones: General: • Código del cliente • Nombre oficial de la Entidad • Siglas • Fecha de creación • Organismo • OSDE/Grupo/Unión • Empresa • Código REEUP u ONE • NIT • Registro Comercial • Dirección • Municipio	

- Provincia

Contactos:

- Nombre y apellidos
- Carnet de identidad
- Email
- Teléfono

Datos bancarios

- Cuenta bancaria en CUP
- Nombre de la cuenta
- Entidad bancaria

Tener la opción “Guardar”.

Tabla A1-6 Historia de usuario “HU08 – Listar cliente”

Fuente: Elaboración propia.

HU08 – Listar cliente	
Yo como:	Usuario del sistema
Quiero:	Ver una lista con todos los Clientes del sistema
Para:	Ver una breve descripción de los Clientes del sistema
Criterios de aceptación • Ver en forma de lista a los Clientes del sistema y sus datos personales principales. • Tener la opción de “Editar” por cada cliente de la lista.	

Tabla A1-7 Historia de usuario “HU09 – Añadir Tipo de Servicio”

Fuente: Elaboración propia.

HU09 – Añadir Tipo de Servicio	
Yo como:	Usuario del sistema

Quiero:	Añadir un Tipo de Servicio nuevo al sistema
Para:	Manejar una nueva clasificación de los servicios
Criterios de aceptación • Completar el campo obligatorio nombre. • Completar el campo descripción. • Tener la opción “Guardar”.	

Tabla A1-8 Historia de usuario “HU10 – Editar Tipo de Servicio”

Fuente: Elaboración propia.

HU10 – Editar Tipo de Servicio	
Yo como:	Usuario del sistema
Quiero:	Editar un Tipo de Servicio en el sistema
Para:	Manejar una nueva clasificación de los servicios
Criterios de aceptación • Modificar el campo nombre. • Modificar el campo descripción. • Tener la opción “Guardar”.	

Tabla A1-9 Historia de usuario “HU11 – Listar Tipo de Servicio”

Fuente: Elaboración propia.

HU11 – Listar Tipo de Servicio	
Yo como:	Usuario del sistema
Quiero:	Ver una lista con todos los tipos de servicios del sistema

Para:	Ver una breve descripción de los tipos de servicios del sistema
Criterios de aceptación • Ver en forma de lista todos los tipos de servicios del sistema. • Tener la opción de “Editar” por cada tipo de servicio de la lista.	

Tabla A1-10 Historia de usuario “HU12 – Añadir Servicio”

Fuente: Elaboración propia.

HU12 – Añadir Servicio	
Yo como:	Usuario del sistema
Quiero:	Añadir un Tipo de servicio nuevo al sistema
Para:	Manejar una nueva clasificación de los servicios
Criterios de aceptación • Seleccionar uno de los tipos de servicios que fueron añadidos. • Completar el campo obligatorio de descripción, donde será insertado el servicio específico que será brindado. • Completar el campo Unidad de medida. • Completar el campo Precio. • Seleccionar la Frecuencia de facturación (Puntual, Mensual, Trimestral, Semestral o Anual). • Tener la opción “Guardar”.	

Tabla A1-11 Historia de usuario “HU13 – Editar Servicio”

Fuente: Elaboración propia.

HU13 – Editar Servicio	
Yo como:	Usuario del sistema
Quiero:	Editar un servicio en el sistema

Para:	Modificar los datos de ese servicio en el sistema
Criterios de aceptación • Seleccionar uno de los tipos de servicios que fueron añadidos. • Completar el campo obligatorio de descripción, donde será insertado el servicio específico que será brindado. • Completar el campo Unidad de medida. • Completar el campo Precio. • Seleccionar la Frecuencia de facturación (Puntual, Mensual, Trimestral, Semestral o Anual). • Tener la opción “Guardar”.	

Tabla A1-12 Historia de usuario “HU14 – Listar Servicio”

Fuente: Elaboración propia.

HU14 – Listar Servicio	
Yo como:	Usuario del sistema
Quiero:	Ver una lista con todos los servicios del sistema
Para:	Ver una breve descripción de los servicios del sistema
Criterios de aceptación • Escoger el Tipo de servicio para ver en forma de lista los servicios que contiene. • Ver en forma de lista todos los servicios del sistema. • Tener la opción de “Editar” por cada servicio de la lista.	

Tabla A1-13 Historia de usuario “HU15 – Añadir Solicitud de Servicio”

Fuente: Elaboración propia.

HU15 – Añadir Solicitud de Servicio	
Yo como:	Usuario del sistema
Quiero:	Añadir una Solicitud de servicio nuevo al sistema

Para:	Manejar una nueva Solicitud de los servicios en el sistema
Criterios de aceptación ● Seleccionar la Fecha de la solicitud del servicio, este campo es obligatorio. ● Seleccionar la Entidad solicitante del servicio (el cliente que ya fue añadido anteriormente al sistema), este campo es obligatorio. ● Seleccionar el Estado de la solicitud del servicio (En evaluación por el área correspondiente, No aceptada por GET, Pendiente presentación oferta o contrato, En proceso de negociación, No aceptada por el cliente o Contrato firmado), este campo es obligatorio. ● Completar los Datos de contacto: ○ Nombre completo, este campo es obligatorio. ○ Cargo, este campo es obligatorio. ○ Correo ○ Teléfono ● Completar el campo de texto Descripción de las necesidades identificadas. ● Tener la opción “Guardar”.	

Tabla A1-14 Historia de usuario “HU16 – Editar Solicitud de Servicio”

Fuente: Elaboración propia.

HU16 – Editar Solicitud de Servicio	
Yo como:	Usuario del sistema
Quiero:	Editar una Solicitud de servicio en el sistema
Para:	Modificar los datos de esa Solicitud servicio en el sistema
Criterios de aceptación ● Seleccionar la Fecha de la solicitud del servicio, este campo es obligatorio. ● Seleccionar la Entidad solicitante del servicio (el cliente que ya fue añadido anteriormente al sistema), este campo es obligatorio.	

- Seleccionar el Estado de la solicitud del servicio (En evaluación por el área correspondiente, No aceptada por GET, Pendiente presentación oferta o contrato, En proceso de negociación, No aceptada por el cliente o Contrato firmado), este campo es obligatorio.
- Completar los Datos de contacto:
 - Nombre completo, este campo es obligatorio.
 - Cargo, este campo es obligatorio.
 - Correo
 - Teléfono
- Completar el campo de texto Descripción de las necesidades identificadas.
- Tener la opción “Guardar”.

Tabla A1-15 Historia de usuario “HU17 – Listar Solicitud de Servicio”

Fuente: Elaboración propia.

HU17 – Listar Solicitud de Servicio	
Yo como:	Usuario del sistema
Quiero:	Ver una lista con todas las Solicitudes de servicios del sistema
Para:	Ver una breve descripción de las Solicitudes de los servicios del sistema
Criterios de aceptación ● Ver en forma de lista todas las solicitudes de servicios del sistema. ● Tener la opción de “Editar” por cada cliente de la lista. ● Tener la opción de “Crear oferta comercial” a partir de una solicitud de servicios de la lista.	

Tabla A1-16 Historia de usuario “HU18 – Crear Oferta Comercial”

Fuente: Elaboración propia.

HU18 – Crear Oferta Comercial	
Yo como:	Usuario del sistema
Quiero:	Crear Oferta Comercial en el sistema

Para:	Poder controlar a través del sistema los servicios que desea recibir el cliente mediante la Oferta comercial
Criterios de aceptación ● Seleccionar la Forma de pago (Efectivo, Transferencia o Mixto). ● Tener la opción “Añadir servicio”. ● Seleccionar de los servicios ya insertados cual desea recibir el cliente. ● Especificar la cantidad de servicio del mismo tipo que desea recibir. ● Especificar el precio del servicio. ● Tener la opción “Eliminar”. ● Completar el campo de texto “Requerimientos mínimos”. ● Tener la opción “Guardar”.	

Tabla A1-17 Historia de usuario “HU19 – Editar Oferta Comercial”

Fuente: Elaboración propia.

HU19 – Editar Oferta Comercial	
Yo como:	Usuario del sistema
Quiero:	Editar una Oferta Comercial en el sistema
Para:	Modificar los datos de esa Oferta Comercial en el sistema
Criterios de aceptación ● Seleccionar la Entidad solicitante del servicio (el cliente que ya fue añadido anteriormente al sistema), este campo es obligatorio. ● Seleccionar la Forma de pago (Efectivo, Transferencia o Mixto). ● Tener la opción “Añadir servicio”. ● Seleccionar de los servicios ya insertados cual desea recibir el cliente. ● Especificar la cantidad de servicio del mismo tipo que desea recibir. ● Especificar el precio del servicio. ● Tener la opción “Eliminar”. ● Completar el campo de texto “Requerimientos mínimos”.	

- Tener la opción “Guardar”.

Tabla A1-18 Historia de usuario “HU20 – Listar Oferta Comercial”

Fuente: Elaboración propia.

HU20 – Listar Oferta Comercial	
Yo como:	Usuario del sistema
Quiero:	Ver una lista con todas las Ofertas Comerciales del sistema
Para:	Ver una breve descripción de las Ofertas Comerciales del sistema
Criterios de aceptación • Ver en forma de lista todas las Ofertas Comerciales del sistema. • Tener la opción de “Editar” por cada Ofertas Comerciales de la lista. • Tener la opción de “Imprimir”.	

Tabla A1-19 Historia de usuario “HU21 – Imprimir Oferta Comercial”

Fuente: Elaboración propia.

HU21 – Imprimir Oferta Comercial	
Yo como:	Usuario del sistema
Quiero:	Imprimir una Oferta Comercial del sistema
Para:	Poder entregarla al cliente de forma física y que este la pueda revisar
Criterios de aceptación • Observar cómo queda la estructura del documento. • Tener la opción de “Imprimir”.	

Tabla A1-20 Historia de usuario “HU22 – Crear Contrato”

Fuente: Elaboración propia.

HU22 – Crear Contrato	
Yo como:	Usuario del sistema
Quiero:	Crear Oferta Comercial en el sistema
Para:	Poder controlar a través del sistema los servicios que desea recibir el cliente mediante la Oferta comercial
Criterios de aceptación ● Seleccionar la Entidad solicitante del servicio (el cliente que ya fue añadido anteriormente al sistema), este campo es obligatorio. ● Especificar el Valor del contrato, este campo es obligatorio. ● Especificar el Tipo de contrato (Contrato de prestación de servicios, Contrato de Comodato, Documentación de Proyectos, Ejecución de obras o Contrato de Arrendamiento), este campo es obligatorio. ● Seleccionar el Anticipo pactado. ● Seleccionar la Forma de pago (Efectivo, Transferencia o Mixto), este campo es obligatorio. ● Especificar la Fecha de firma del contrato, este campo es obligatorio. ● Especificar la fecha de Vigencia del contrato, este campo es obligatorio. ● Tener la opción CL que Incluye traspaso de CL sobre el valor contratado. ● Completar el campo de texto “Identificación de las partes”, este campo es obligatorio. ● Completar el campo de texto “Objeto del contrato”, este campo es obligatorio. ● Completar el campo de texto “Dictamen Legal”, este campo es obligatorio. ● Completar el campo de texto “Acuerdo del CA o CD”. ● Completar el campo de texto “Observaciones”. ● Tener la opción “Guardar” para guardarlo como proforma de contrato. ● Tener la opción “Crear contrato” establecer el contrato como firmado.	

Tabla A1-21 Historia de usuario “HU23 – Listar Contrato”**Fuente: Elaboración propia.**

HU23 – Listar Contrato	
Yo como:	Usuario del sistema
Quiero:	Ver una lista con todos los Contratos del sistema
Para:	Ver una breve descripción de las Ofertas Comerciales del sistema
Criterios de aceptación • Ver en forma de lista todos los Contratos del sistema. • Tener la opción de “Editar” por cada Contrato de la lista. • Tener la opción de “Imprimir”.	

Tabla A1-22 Historia de usuario “HU24 – Imprimir Contrato”**Fuente: Elaboración propia.**

HU24 – Imprimir Contrato	
Yo como:	Usuario del sistema
Quiero:	Imprimir un Contrato del sistema
Para:	Poder entregarla al cliente de forma física y que este la pueda revisar y firmar
Criterios de aceptación • Observar cómo queda la estructura del documento. • Tener la opción de “Imprimir”.	

Tabla A1-23 Historia de usuario “HU25 – Crear Suplemento”**Fuente: Elaboración propia.**

HU25 – Crear Suplemento	
Yo como:	Usuario del sistema

Quiero:	Crear un Suplemento en el sistema
Para:	Poder realizar alguna modificación dentro de un contrato del sistema
Criterios de aceptación • En la lista de contratos, seleccionar el contrato que tenga como Estado “Firmado” y presionar el botón de Editar para crear un suplemento de ese contrato. • Seleccionar la Entidad solicitante del servicio (el cliente que ya fue añadido anteriormente al sistema), este campo es obligatorio. • Especificar el Valor del contrato, este campo es obligatorio. • Especificar el Tipo de contrato (Contrato de prestación de servicios, Contrato de Comodato, Documentación de Proyectos, Ejecución de obras o Contrato de Arrendamiento), este campo es obligatorio. • Seleccionar el Anticipo pactado. • Seleccionar la Forma de pago (Efectivo, Transferencia o Mixto), este campo es obligatorio. • Especificar la Fecha de firma del contrato, este campo es obligatorio. • Especificar la fecha de Vigencia del contrato, este campo es obligatorio. • Tener la opción CL que Incluye traspaso de CL sobre el valor contratado. • Completar el campo de texto “Identificación de las partes”, este campo es obligatorio. • Completar el campo de texto “Objeto del contrato”, este campo es obligatorio. • Completar el campo de texto “Dictamen Legal”, este campo es obligatorio. • Completar el campo de texto “Acuerdo del CA o CD”. • Completar el campo de texto “Observaciones”. • Tener la opción “Guardar”.	

Tabla A1-24 Historia de usuario “HU26 – Listar Suplemento”

Fuente: Elaboración propia.

HU26 – Listar Suplemento	
Yo como:	Usuario del sistema
Quiero:	Ver una lista con todos los Suplementos del contrato seleccionado

Para:	Ver una breve descripción de los Suplementos del sistema
Criterios de aceptación • Ver en forma de lista todos los Suplementos que se han realizado a partir de un contrato.	

Tabla A1-25 Historia de usuario “HU27 – Añadir Reporte de Producción”

Fuente: Elaboración propia.

HU27 – Añadir Reporte de Producción	
Yo como:	Usuario del sistema
Quiero:	Añadir una Solicitud de servicio nuevo al sistema
Para:	Manejar una nueva Solicitud de los servicios en el sistema
Criterios de aceptación • Seleccionar el área o brigada que brinda los servicios, este campo es obligatorio. • Seleccionar Mes al que pertenece el Reporte de Producción, este campo es obligatorio. • Seleccionar el Plan de ingresos: El plan aprobado para el período que se está facturando, este campo es obligatorio. • Especificar el Código de cliente: El número que identifica al cliente en el sistema informático, este campo es obligatorio. • Seleccionar la Entidad solicitante del servicio (el cliente que ya fue añadido anteriormente al sistema), este campo es obligatorio, este campo es obligatorio. • Especificar el No. de Contrato: Contrato que ampara la facturación, este campo es obligatorio. • Seleccionar de los servicios ya insertados cual o cuales se va a facturar, este campo es obligatorio. • Especificar la Cantidad de los servicios a facturar, este campo es obligatorio. • Especificar el importe total del servicio, de acuerdo a la cantidad a facturar, este campo es obligatorio. • Especificar el centro de costo que corresponde, este campo es obligatorio.	

- Seleccionar la modalidad: Iguala o puntual.
- Seleccionar el Periodo a Facturar (Mes, trimestre, semestre, año).
- Especificar el Número de factura generada correspondiente al importe a cobrar.
- Especificar el importe total que se va a facturar en el mes
- Especificar Confeccionado por: Persona que elaboró el Reporte, este campo es obligatorio.
- Especificar Conciliado con: Personal designado con que se concilió, este campo es obligatorio.
- Especificar Aprobado por: Director de UEB o persona designada para la aprobación, este campo es obligatorio.
- Tener la opción “Guardar”.

Tabla A1-26 Historia de usuario “HU28 – Editar Reporte de Producción”

Fuente: Elaboración propia.

HU28 – Editar Reporte de Producción	
Yo como:	Usuario del sistema
Quiero:	Editar un Reporte de Producción en el sistema
Para:	Modificar los datos de ese Reporte de Producción en el sistema
Criterios de aceptación • Seleccionar el área o brigada que brinda los servicios, este campo es obligatorio. • Seleccionar Mes al que pertenece el Reporte de Producción, este campo es obligatorio. • Seleccionar el Plan de ingresos: El plan aprobado para el período que se está facturando, este campo es obligatorio. • Especificar el Código de cliente: El número que identifica al cliente en el sistema informático, este campo es obligatorio. • Seleccionar la Entidad solicitante del servicio (el cliente que ya fue añadido anteriormente al sistema), este campo es obligatorio, este campo es obligatorio. • Especificar el No. de Contrato: Contrato que ampara la facturación, este campo es obligatorio. • Seleccionar de los servicios ya insertados cual o cuales se va a facturar, este campo es obligatorio.	

- Especificar la Cantidad de los servicios a facturar, este campo es obligatorio.
- Especificar el importe total del servicio, de acuerdo a la cantidad a facturar, este campo es obligatorio.
- Especificar el centro de costo que corresponde, este campo es obligatorio.
- Seleccionar la modalidad: Iguala o puntual.
- Seleccionar el Periodo a Facturar (Mes, trimestre, semestre, año).
- Especificar el Número de factura generada correspondiente al importe a cobrar.
- Especificar el importe total que se va a facturar en el mes
- Especificar Confeccionado por: Persona que elaboró el Reporte, este campo es obligatorio.
- Especificar Conciliado con: Personal designado con que se concilió, este campo es obligatorio.
- Especificar Aprobado por: Director de UEB o persona designada para la aprobación, este campo es obligatorio.
- Tener la opción “Guardar”.

Tabla A1-27 Historia de usuario “HU29 – Listar Reporte de Producción”

Fuente: Elaboración propia.

HU29 – Listar Reporte de Producción	
Yo como:	Usuario del sistema
Quiero:	Ver una lista con todos los Reportes de Producción del sistema
Para:	Ver una breve descripción de los Reporte de Producción del sistema
Criterios de aceptación • Ver en forma de lista todos los Reporte de Producción del sistema. • Tener la opción de “Editar” por cada Suplemento de la lista. • Tener la opción de “Imprimir”.	

Tabla A1-28 Historia de usuario “HU30 – Imprimir Reporte de Producción”

Fuente: Elaboración propia.

HU30 – Imprimir Reporte de Producción

Yo como:	Usuario del sistema
Quiero:	Imprimir Reporte de Producción del sistema
Para:	Reunir la información de los servicios que fueron brindados por cada departamento de la UEB y así proceder a la facturación
Criterios de aceptación • Observar cómo queda la estructura del documento. • Tener la opción de “Imprimir”.	

Tabla A1-29 Historia de usuario “HU31 – Imprimir Factura”

Fuente: Elaboración propia.

HU31 – Imprimir Factura	
Yo como:	Usuario del sistema
Quiero:	Imprimir Factura del sistema
Para:	Poder llevarla o enviársela por correo al cliente y que este proceda a firmarla
Criterios de aceptación • Observar cómo queda la estructura del documento. • Tener la opción de “Imprimir”.	

Tabla A1-30 Historia de usuario “HU32 – Generar el Contrato a imprimir”

Fuente: Elaboración propia.

HU32 – Generar el Contrato a imprimir	
Yo como:	Usuario del sistema
Quiero:	Generar un Contrato del sistema
Para:	Poder imprimirlo y entregárselo al cliente y que este proceda a firmarlo

Criterios de aceptación • Visualizar la estructura final del documento del contrato. • Tener la opción de “Imprimir”.

Tabla A1-31 Historia de usuario “HU33 – Generar la Factura a imprimir”

Fuente: Elaboración propia.

HU33 – Generar la Factura a imprimir	
Yo como:	Usuario del sistema
Quiero:	Generar la Factura del sistema
Para:	Poder imprimir la Factura y que pueda ser firmada por el cliente
Criterios de aceptación • Visualizar la estructura final del documento de la factura. • Tener la opción de “Imprimir”.	

Anexo 2: Pruebas de aceptación

Tabla A2-1 Pruebas de aceptación “PA2 – Registrar usuario”

Fuente: Elaboración propia.

ID	PA2
Historia de usuario	HU02 - Registrar usuario
Aprobado / ID del defecto:	Aprobado
Descripción • El Administrador del sistema debe poder introducir las credenciales del usuario. ○ Nombre, campo obligatorio. ○ Apellidos, campo obligatorio. ○ Correo, campo obligatorio. ○ Seleccionar el estado, “Activo”, “Inactivo” o si está “Eliminado”.	

○ Teléfono ○ Dirección ○ Contraseña, campo obligatorio. ● Al dejar campos en blanco el sistema debe mostrar el mensaje de “Este campo no puede estar vacío” debajo de los respectivos campos en blanco. ● Al insertar un correo no válido el sistema debe mostrar el mensaje de “El correo no es válido”. ● En caso de que se estén insertando menos de 6 caracteres en la contraseña el sistema debe mostrar el mensaje de “Este campo debe contener al menos 6 caracteres”. ● Si no existe ningún problema al presionar el botón “Guardar” el administrador debe poder registrar al usuario en el sistema.
Resultado esperado 1. El usuario se haya creado correctamente 2. El usuario debe poder ingresar al sistema con sus credenciales nuevas

Tabla A2-2 Pruebas de aceptación “PA3 – Editar usuario”

Fuente: Elaboración propia.

ID	PA3
Historia de usuario	HU03 – Editar usuario
Aprobado / ID del defecto:	Aprobado
Descripción ● El Administrador del sistema debe poder realizar cambios en las credenciales del usuario. ○ Nombre, campo obligatorio. ○ Apellidos, campo obligatorio. ○ Correo, campo obligatorio. ○ Seleccionar el estado, “Activo”, “Inactivo” o si está “Eliminado”. ○ Teléfono ○ Dirección Cambiar contraseña ○ Contraseña, campo obligatorio. ○ Repetir la contraseña para comprobar que es la correcta.	

- Al dejar campos en blanco el sistema debe mostrar el mensaje de “Este campo no puede estar vacío” debajo de los respectivos campos en blanco.
- Al insertar un correo no válido el sistema debe mostrar el mensaje de “El correo no es válido”.
- En caso de que se estén insertando menos de 6 caracteres en la contraseña el sistema debe mostrar el mensaje de “Este campo debe contener al menos 6 caracteres”.
- Si no existe ningún problema al presionar el botón “Guardar” el administrador debe poder salvar los cambios realizados en la cuenta del usuario en el sistema.

Resultado esperado

1. Se hayan realizado correctamente los cambios en la cuenta del usuario.
2. El usuario debe poder ingresar al sistema con sus credenciales modificadas.

Tabla A2-3 Pruebas de aceptación “PA4 – Listar usuario”

Fuente: Elaboración propia.

ID	PA4
Historia de usuario	HU04 – Listar usuario
Aprobado / ID del defecto:	Aprobado
Descripción - El Administrador del sistema deberá poder ver los datos personales del usuario. - Nombre - Apellidos - Correo - Teléfono - Estado, “Activo”, “Inactivo” o si está “Eliminado”. - Y podrá seleccionar la opción “Editar usuario”	
Resultado esperado Poder visualizar una lista con todos los usuarios registrados en el sistema.	

Tabla A2-4 Pruebas de aceptación “PA5 – Deshabilitar usuario”

Fuente: Elaboración propia.

ID	PA5
Historia de usuario	HU05 – Deshabilitar Usuario
Aprobado / ID del defecto:	Aprobado
Descripción En el botón “Editar usuario” en el campo “Estado” se deberá seleccionar la opción “Eliminado” para quitarle el acceso al sistema a algún usuario.	
Resultado esperado El usuario al que se le quitó el acceso no podrá acceder al sistema.	

Tabla A2-5 Pruebas de aceptación "PA6 – Añadir usuario"

Fuente: Elaboración propia.

ID	PA6
Historia de usuario	HU06 – Añadir Cliente
Aprobado / ID del defecto:	Aprobado
Descripción Todos los campos listados en la historia de usuario "HU06 – Añadir Cliente" estén presentes en el sistema.	
Resultado esperado El usuario podrá añadir clientes al sistema.	

Tabla A2-6 Pruebas de aceptación "PA7 – Editar Cliente"

Fuente: Elaboración propia.

ID	PA7
Historia de usuario	HU07 – Editar Cliente
Aprobado / ID del defecto:	Aprobado
Descripción Todos los campos listados en la historia de usuario "HU07 – Editar Cliente" estén presentes en el sistema y se podrán realizar modificaciones en los campos que contengan información.	
Resultado esperado - El usuario podrá editar los datos del cliente en el sistema. - Debe poder brindar la opción de Guardar los datos modificados del cliente.	

Tabla A2-7 Pruebas de aceptación "PA8 – Listar cliente"

Fuente: Elaboración propia.

ID	PA8
Historia de usuario	HU08 – Listar cliente
Aprobado / ID del defecto:	Aprobado
Descripción	

- El usuario podrá ver en forma de lista todos los clientes que hayan sido introducidos al sistema.

Resultado esperado

1. Poder visualizar una lista con todos los clientes introducidos en el sistema.
2. Debe poder brindar la opción de editar el cliente deseado.

Tabla A2-8 Pruebas de aceptación “PA9 – Añadir Tipo de Servicio”*Fuente: Elaboración propia.*

ID	PA9
Historia de usuario	HU09 – Añadir Tipo de Servicio
Aprobado / ID del defecto:	Aprobado
Descripción • Todos los campos listados en la historia de usuario “HU09 – Añadir Tipo de Servicio” estén presentes en el sistema.	
Resultado esperado 1. El usuario podrá añadir un Tipo de Servicio nuevo al sistema.	

Tabla A2-9 Pruebas de aceptación “PA10 – Editar Tipo de Servicio”*Fuente: Elaboración propia.*

ID	PA10
Historia de usuario	HU10 – Editar Tipo de Servicio
Aprobado / ID del defecto:	Aprobado
Descripción • Todos los campos listados en la historia de usuario “HU10 – Editar Tipo de Servicio” estén presentes en el sistema y se podrán realizar modificaciones en los campos que contengan información.	
Resultado esperado 1. El usuario podrá editar los datos del Tipo de Servicio en el sistema.	

Tabla A2-10 Pruebas de aceptación “PA11 – Listar Tipo de Servicio”

Fuente: Elaboración propia.

ID	PA11
Historia de usuario	HU11 – Listar Tipo de Servicio
Aprobado / ID del defecto:	Aprobado
Descripción El usuario podrá ver en forma de lista los Tipos de servicios que hayan sido introducidos al sistema.	
Resultado esperado - Poder visualizar una lista con todos los Tipos de servicios introducidos en el sistema. - Debe poder brindar la opción de editar el Tipo de servicio deseado.	

Tabla A2-11 Pruebas de aceptación “PA12 – Añadir Servicio”

Fuente: Elaboración propia.

ID	PA12
Historia de usuario	HU12 – Añadir Servicio
Aprobado / ID del defecto:	Aprobado
Descripción Todos los campos listados en la historia de usuario “HU12 – Añadir Servicio” estén presentes en el sistema.	
Resultado esperado - El usuario podrá añadir un Servicio nuevo al sistema a partir de un Tipo de servicio. - Debe poder brindar la opción de Guardar servicio.	

Tabla A2-12 Pruebas de aceptación “PA13 – Editar Servicio”

Fuente: Elaboración propia.

ID	PA13
Historia de usuario	HU13 – Editar Servicio
Aprobado / ID del defecto:	Aprobado

Descripción Todos los campos listados en la historia de usuario “HU12 – Añadir Servicio” estén presentes en el sistema y poder realizar cambios en cada campo.
Resultado esperado - El usuario podrá editar un Servicio en el sistema. - Debe poder brindar la opción de Guardar servicio.

Tabla A2-13 Pruebas de aceptación “PA14 – Listar Servicio”

Fuente: Elaboración propia.

ID	PA14
Historia de usuario	HU14 – Listar Servicio
Aprobado / ID del defecto:	Aprobado
Descripción El usuario podrá ver en forma de lista los Servicios que hayan sido introducidos al sistema.	
Resultado esperado - Poder visualizar una lista con todos los Servicios en sus respectivos Tipos de servicios introducidos en el sistema. - Debe poder brindar la opción de editar el Tipo de servicio deseado.	

Tabla A2-14 Pruebas de aceptación “PA15 – Añadir Solicitud de Servicio”

Fuente: Elaboración propia.

ID	PA15
Historia de usuario	HU15 – Añadir Solicitud de Servicio
Aprobado / ID del defecto:	Aprobado
Descripción Todos los campos listados en la historia de usuario “HU15 – Añadir Solicitud de Servicio” estén presentes en el sistema.	
Resultado esperado - El usuario podrá añadir una Solicitud de Servicio nueva al sistema - Debe poder brindar la opción de Guardar la Solicitud de servicio insertada en el sistema.	

Tabla A2-15 Pruebas de aceptación “PA16 – Editar Solicitud de Servicio”

Fuente: Elaboración propia.

ID	PA16
Historia de usuario	HU16 – Editar Solicitud de Servicio
Aprobado / ID del defecto:	Aprobado
Descripción Todos los campos listados en la historia de usuario “HU16 – Editar Solicitud de Servicio” estén presentes en el sistema y poder realizar cambios en cada campo.	
Resultado esperado - El usuario podrá editar una Solicitud de Servicio en el sistema. - Debe poder brindar la opción de Guardar Solicitud de servicio insertada en el sistema.	

Tabla A2-16 Pruebas de aceptación “PA17 – Listar Solicitud de Servicio”

Fuente: Elaboración propia.

ID	PA17
Historia de usuario	HU17 – Listar Solicitud de Servicio
Aprobado / ID del defecto:	Aprobado
Descripción El usuario podrá ver en forma de lista las Solicitudes de Servicios que hayan sido introducidos al sistema.	
Resultado esperado - Poder visualizar una lista con todas las Solicitudes de Servicios introducidas en el sistema. - Debe poder brindar la opción de editar el Tipo de servicio deseado. - Debe poder brindar la opción de Crear oferta Comercial.	

Tabla A2-17 Pruebas de aceptación “PA18 – Crear Oferta Comercial”

Fuente: Elaboración propia.

ID	PA18
Historia de usuario	HU18 – Crear Oferta Comercial

Aprobado / ID del defecto:	Aprobado
Descripción Todos los campos listados en la historia de usuario “HU18 – Crear Oferta Comercial” estén presentes en el sistema.	
Resultado esperado - El usuario podrá crear una Oferta Comercial nueva en el sistema. - Debe poder brindar la opción de Guardar la Solicitud de servicio insertada en el sistema.	

Tabla A2-18 Pruebas de aceptación “PA19 – Editar Oferta Comercial”

Fuente: Elaboración propia.

ID	PA19
Historia de usuario	HU19 – Editar Oferta Comercial
Aprobado / ID del defecto:	Aprobado
Descripción Todos los campos listados en la historia de usuario “HU19 – Editar Oferta Comercial” estén presentes en el sistema y poder realizar cambios en cada campo.	
Resultado esperado - El usuario podrá editar una Oferta Comercial en el sistema. - Debe poder brindar la opción de Guardar Solicitud de servicio insertada en el sistema.	

Tabla A2-19 Pruebas de aceptación “PA20 – Listar Oferta Comercial”

Fuente: Elaboración propia.

ID	PA20
Historia de usuario	HU20 – Listar Oferta Comercial
Aprobado / ID del defecto:	Aprobado
Descripción El usuario podrá ver en forma de lista las Ofertas Comerciales que hayan sido introducidas en el sistema.	
Resultado esperado Poder visualizar una lista con todas las Ofertas Comerciales introducidas en el sistema.	

2. Debe poder brindar la opción de editar la Oferta Comercial deseada.
3. Debe poder brindar la opción de Imprimir la Oferta Comercial.

Tabla A2-20 Pruebas de aceptación “PA21 – Imprimir Oferta Comercial”

Fuente: Elaboración propia.

ID	PA21
Historia de usuario	HU21 – Imprimir Oferta Comercial
Aprobado / ID del defecto:	Aprobado
Descripción El usuario podrá Imprimir las Ofertas Comerciales que hayan sido introducidas en el sistema.	
Resultado esperado Debe poder Imprimir la Oferta Comercial.	

Tabla A2-21 Pruebas de aceptación “PA22 – Crear Contrato”

Fuente: Elaboración propia.

ID	PA22
Historia de usuario	HU22 – Crear Contrato
Aprobado / ID del defecto:	Aprobado
Descripción Todos los campos listados en la historia de usuario “HU22 – Crear Contrato” estén presentes en el sistema.	
Resultado esperado - El usuario podrá crear un Contrato nuevo en el sistema. - Debe poder brindar la opción de Guardar el Contrato insertado en el sistema como proforma de contrato o como contrato firmado.	

Tabla A2-22 Pruebas de aceptación “PA23 – Listar Contrato”

Fuente: Elaboración propia.

ID	PA23
Historia de usuario	HU23 – Listar Contrato
Aprobado / ID del defecto:	Aprobado
Descripción El usuario podrá ver en forma de lista los Contratos que hayan sido introducidos en el sistema.	
Resultado esperado - Poder visualizar una lista con todos los Contratos introducidos en el sistema. - Debe poder brindar la opción de editar el Contrato deseado. - Debe poder brindar la opción de Imprimir el Contrato deseado.	

Tabla A2-23 Pruebas de aceptación “PA24 – Imprimir Contrato”

Fuente: Elaboración propia.

ID	PA24
Historia de usuario	HU24 – Imprimir Contrato
Aprobado / ID del defecto:	Aprobado
Descripción El usuario podrá Imprimir los Contratos que hayan sido introducidos en el sistema.	
Resultado esperado Debe poder Imprimir el Contrato deseado.	

Tabla A2-24 Pruebas de aceptación “PA25 – Crear Suplemento”

Fuente: Elaboración propia.

ID	PA25
Historia de usuario	HU25 – Crear Suplemento
Aprobado / ID del defecto:	Aprobado
Descripción	

- Todos los campos listados en la historia de usuario “HU25 – Crear Suplemento” estén presentes en el sistema.

Resultado esperado

1. El usuario podrá crear un Suplemento nuevo en el sistema al Contrato que ya está previamente firmado.
2. Debe poder brindar la opción de Guardar los cambios realizados al contrato como un Suplemento nuevo.

Tabla A2-25 Pruebas de aceptación “PA26 – Listar Suplemento”*Fuente: Elaboración propia.*

ID	PA26
Historia de usuario	HU26 – Listar Suplemento
Aprobado / ID del defecto:	Aprobado
Descripción • El usuario podrá ver en forma de lista dentro del contrato que se está modificando los Suplementos que hayan sido introducidos en el sistema.	
Resultado esperado 1. Poder visualizar una lista con todos los Suplementos que se le hayan realizado a un contrato en específico. 2. Debe poder brindar la opción de Guardar el Suplemento.	

Tabla A2-26 Pruebas de aceptación “PA27 – Añadir Reporte de Producción”*Fuente: Elaboración propia.*

ID	PA27
Historia de usuario	HU27 – Añadir Reporte de Producción
Aprobado / ID del defecto:	Aprobado
Descripción • Todos los campos listados en la historia de usuario “HU27 – Añadir Reporte de Producción” estén presentes en el sistema.	
Resultado esperado 1. El usuario podrá añadir un Reporte de Producción nuevo en el sistema.	

2. Debe poder brindar la opción de Guardar el Reporte de Producción insertado en el sistema.

Tabla A2-27 Pruebas de aceptación “PA28 – Editar Reporte de Producción”

Fuente: Elaboración propia.

ID	PA28
Historia de usuario	HU28 – Editar Reporte de Producción
Aprobado / ID del defecto:	Aprobado
Descripción Todos los campos listados en la historia de usuario “HU28 – Editar Reporte de Producción” estén presentes en el sistema.	
Resultado esperado - El usuario podrá editar un Reporte de Producción al sistema. - Debe poder brindar la opción de Guardar el Reporte de Producción insertado en el sistema.	

Tabla A2-28 Pruebas de aceptación “PA29 – Listar Reporte de Producción”

Fuente: Elaboración propia.

ID	PA29
Historia de usuario	HU29 – Listar Reporte de Producción
Aprobado / ID del defecto:	Aprobado
Descripción El usuario podrá ver en forma de lista los Reportes de Producción que hayan sido introducidos al sistema.	
Resultado esperado - Poder visualizar una lista con todos los Reportes de Producción introducidos en el sistema. - Debe poder brindar la opción de editar el Reporte de Producción deseado.	

Tabla A2-29 Pruebas de aceptación “PA30 – Imprimir Reporte de Producción”

Fuente: Elaboración propia.

ID	PA30
Historia de usuario	HU30 – Imprimir Reporte de Producción
Aprobado / ID del defecto:	Aprobado
Descripción El usuario podrá Imprimir los Reportes de Producción que hayan sido introducidos en el sistema.	
Resultado esperado Debe poder Imprimir el Reporte de Producción deseado.	

Tabla A2-30 Pruebas de aceptación “PA31 – Imprimir Factura”

Fuente: Elaboración propia.

ID	PA31
Historia de usuario	HU31 – Imprimir Factura
Aprobado / ID del defecto:	Aprobado
Descripción El usuario podrá Imprimir las Factura que hayan sido introducidos en el sistema.	
Resultado esperado Debe poder Imprimir la Factura deseada.	

Tabla A2-31 Pruebas de aceptación “PA32 – Generar el Contrato a imprimir”

Fuente: Elaboración propia.

ID	PA32
Historia de usuario	HU32 – Generar el Contrato a imprimir
Aprobado / ID del defecto:	Aprobado
Descripción Mostrar una vista de cómo está la estructura del contrato.	

Resultado esperado

1. El usuario podrá ver el resultado final del documento
2. Tener una opción de imprimir el contrato

Tabla A2-32 Pruebas de aceptación “PA33 – Generar la Factura a imprimir”

Fuente: Elaboración propia.

ID	PA33
Historia de usuario	HU33 – Generar la Factura a imprimir
Aprobado / ID del defecto:	Aprobado
Descripción • Todos los campos listados en la historia de usuario “HU25 – Crear Suplemento” estén presentes en el sistema.	
Resultado esperado 1. El usuario podrá crear un Suplemento nuevo en el sistema al Contrato que ya está previamente firmado. 2. Debe poder brindar la opción de Guardar los cambios realizados al contrato como un Suplemento nuevo.	